AD-A051 832 SRI INTERNATIONAL MENLO PARK CA TRANSPORTATION AND I--ETC F/G 9/2

DEVELOPMENT OF AN INTERACTIVE BALFRAM COMPUTER PROGRAM: PHASE I--ETC(U)

NOV 77 B LAURENCE, P TUAN N00014-76-C-1111

UNCLASSIFIED SRI-TR-5822-1 NL

1 OF 2

AD
A051832

(12)

DEVELOPMENT OF AN INTERACTIVE BALFRAM COMPUTER PROGRAM: PHASE I — MODEL REVISION

Technical Report for the Period 1 October 1976
to 30 November 1977

February 1978

By: Bert Laurence
Paul Tuan

Prepared for:

Commander in Chief Pacific
Research and Analysis Office (J77)
Box 13
Camp Smith, Hawaii 96861

Office of Naval Research
Fleet Analysis and Support Division (Code 230)
800 North Quincy
Arlington, VA 22217

SRI international
333 Ravenswood Avenue
Menlo Park, California 94025
(415) 326-6200
Cable: SRI INTL MPK
TWX: 910-373-1246

SRI International

DEVELOPMENT OF AN INTERACTIVE BALFRAM COMPUTER PROGRAM: PHASE I — MODEL REVISION

Technical Report for the Period 1 October 1976
to 30 November 1977

February 1978

By: Bert Laurence
Paul Tuan

Prepared for:

Commander in Chief Pacific
Research and Analysis Office (J77)
Box 13
Camp Smith, Hawaii 96861

Office of Naval Research
Fleet Analysis and Support Division (Code 230)
800 North Quincy
Arlington, VA 22217

SRI Project No. 5822
Report TR-5822-1

SL B

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER TR-5822-1	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) DEVELOPMENT OF AN INTERACTIVE BALFRAM COMPUTER PROGRAM: Phase I--Model Revision		5. TYPE OF REPORT & PERIOD COVERED Technical Report for the Period 1 Oct 76 to 30 Nov 77
		6. PERFORMING ORG. REPORT NUMBER SRI Project 5822-1
7. AUTHOR(s) B. Laurence, P. Tuan		8. CONTRACT OR GRANT NUMBER(s) N00014-76-C-1111
9. PERFORMING ORGANIZATION NAME AND ADDRESS SRI International, Transportation and Industrial Systems Center, Menlo Park, CA 94025		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NR-364-266
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research, Fleet Analysis and Support Division (Code 230), Arlington, VA 22217	12. REPORT DATE 30 Nov 77	13. NO. OF PAGES 107
14. MONITORING AGENCY NAME & ADDRESS (if diff. from Controlling Office) Commander in Chief Pacific, Research and Analysis Office (J77), Camp Smith, HI 96861		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE Not Applicable

Handwritten annotations: (9) Technical rept. 1 Oct 76-30 Nov 77; (6); (10) Bert/Laurence, Paul/Tuan; (15); new; (11); (12) 104 p.; (14) SRI-TR-5822-1

16. DISTRIBUTION STATEMENT (of this report)

Unlimited distribution

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from report)

Unlimited distribution

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Force capabilities analysis
Force requirements
Lanchester laws of combat
Simulation
Wargaming
Computer programming
WWMCCS

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

BALFRAM is a force planning model with land, sea, and air interactions based on Lanchester law attrition. This technical report describes Phase I of a project to convert the BALFRAM computer program to an interactive program where a model user can manipulate decision parameters to control problem solution during the computer execution. This report presents the technical details of the modifications made during this model revision phase.

410 615

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

PRECEDING PAGE BLANK - NOT FILMED

PREFACE

This technical report describes Phase One of a project to convert the BALFRAM computer program from a batch-oriented program to an interactive program. Interactive means that the decision maker or other model user can communicate directly with the model as it executes on the computer and examine and manipulate decision parameters to directly control the problem solution. This report presents the technical details of the modifications made during this model revision phase; this report is not intended to present a complete description of the operation and use of BALFRAM. For complete documentation on BALFRAM, readers are referred to the following documents:

E. H. Means, C. L. Phillips, and S. E. Young, BALFRAM User Manual for the Staff of the Commander in Chief Pacific, Stanford Research Institute, Menlo Park, CA 94025, Technical Note NWRC-TN-52 (September 1974).

O. F. Forsyth, C. L. Phillips, and S. E. Young, BALFRAM Program Maintenance Manual for the Staff of the Commander in Chief Pacific, Stanford Research Institute, Menlo Park, CA 94025, Technical Note NWRC-TN-53 (December 1974).

E. H. Means, BALFRAM Seminar Guide, Volumes I and II, Stanford Research Institute, Menlo Park, CA 94025, Technical Note NWRC-TN-63 (February 1976).

Preceding Page Blank - Filmed

CONTENTS

Preceding Page Blank - NOT FILMED

ILLUSTRATIONS

TABLES

Preceding Page Blank - NOT FILMED

ACKNOWLEDGMENTS

The authors wish to express their sincere gratitude to the staff of the Commander in Chief Pacific (CINCPAC) for their assistance during this research. In particular thanks are due to Commander Karl Eulenstein, USN, of the Research and Analysis Office (J77) for his contribution to the restructuring of BALFRAM and for providing insight into the use of BALFRAM for analytic studies at CINCPAC, and to Mr. Roy F. Linsenmeyer, Chief of the Research and Analysis Office for his support and guidance throughout the course of this project.

We also wish to express our appreciation for direction and support provided by Mr. Robert Miller, Director of the Fleet Analysis and Support Division, Office of Naval Research and his staff.

1. INTRODUCTION

1.1 BACKGROUND

BALFRAM is a computer model used in planning and analysis studies for investigating military force interactions. The BALFRAM technique uses specially designed input descriptions of military interactions to construct a computer model tailored to a military situation or scenario. These input commands permit the military situation to be described from the perspective of, and in the terminology of, the military planner. Using this technique, a planner can: analyze combinations of land, sea, and air force levels; analyze capabilities, deployments, and employment strategies required to achieve a desired outcome; and estimate the effectiveness of available forces in carrying out given operational plans.

BALFRAM has evolved over the past decade through a research and development program sponsored by the Office of Naval Research (ONR). Partial support for the program has been provided by the Commander in Chief Pacific (CINCPAC) in recognition of BALFRAM's continuing value to the planning function of that headquarters. BALFRAM computer programs have been implemented on the World Wide Military Command and Control System (WWMCCS) computer at CINCPAC. BALFRAM is also in use at: Headquarters, Pacific Air Forces; the U.S.-Taiwan Defense Command; the Ministry of National Defense of the Republic of China; and the U.S. Naval Surface Weapons Center. The BALFRAM methodology is now being transferred to the U.S./ROK Operational Planning Staff, U.S. Forces Korea. In addition BALFRAM has been used in studies performed for the Chief of Naval Operations (OP-60 and OP-96), CINCPACFLT, and Headquarters Marine Corps.

The potential use of BALFRAM by various Naval and other service commands is expected to increase. However, because of the evolutionary nature of the development of BALFRAM methodology and the batch-processing orientation of the model, restructuring work was done during this research

to enhance the basic BALFRAM concept and to make it a more effective and responsive tool for military decision makers.

1.2 THE CONCEPT OF AN INTERACTIVE BALFRAM

The idea of an interactive BALFRAM program has existed for some time. The proposed ultimate BALFRAM program is a CRT-oriented process whereby the decision makers or analysts can interact with the computer dynamically by using on-line terminals.

The conventional method of computer-aided modeling is essentially a batch process, which does not adequately meet the needs of decision makers. All the parameters and decision rules must be specified before the model is run. The user has no knowledge of the interim results while the model is executing and therefore cannot terminate computer processing if his objective has been achieved of if the processing has become unstable. Once computer processing begins, the user cannot change parameters to facilitate convergence of the problem or study alternatives that might become evident during the processing. Such alternatives may not be evident in the masses of printed output currently generated. These disadvantages result in "middlemen" (analysts, programmers, data aides) being interposed between the decision maker and the model. The result is a long reaction time between the user's decision and the model feedback.

In the context of computerized wargaming, "interactive process" connotes a model that is dynamically responsive to the decision maker. It implies that the decision maker can communicate with the executing model during problem solving and options analysis. The decision maker can examine and manipulate various aspects of his decision parameters to control the process of problem convergence.

Because of the shortened cycle time and increased system responsiveness, the decision makers will take greater advantage of an interactive computer-aided model as a decision tool. The added flexibility and feedback capabilities make convergence to an optimal set of solutions more probable. The decision maker gains greater understanding of his problem

and the parameter interrelationships through model manipulation. More subtle and complex judgments that defy computer algorithms can be included to more realistically represent the real world.

1.3 PHASE ONE--MODEL REVISION

ONR and CINCPAC funded this research program for the purpose of developing an interactive BALFRAM. However, current users of BALFRAM (especially CINCPAC) have, over the years since BALFRAM was installed, accumulated requirements for modifications to the batch version of BALFRAM. Because these modifications affect the utility of BALFRAM for operational commands and because interactive BALFRAM could not be fully implemented immediately, this research program was directed toward the needed modifications and other preparations for interactive BALFRAM.

The desired modifications can be divided into two classes. Structural modifications were made to remove superfluous routines that had been superceded and to restructure the remaining routines into new execution modules that would be more compact in memory. The second class of modifications improved the utility of certain descriptors that controlled the progress of the force interactions. New and changed capabilities were desired that would make describing military scenarios and force interactions more efficient.

A second objective of the research described in this report was to improve the documentation of the BALFRAM programs in preparation for the major changes that would take place during the design and implementation of interactive BALFRAM. Changes made to the model software prior to and during this research needed to be adequately documented, and the documentation needed to be brought up to date. This new documentation would be useful not only for the interactive systems design, but also for all the other current users of BALFRAM.

1.4 REPORT ORGANIZATION

The desired model modifications have been made and are described in this report. Section 2 discusses three areas of BALFRAM modifications.

The major modifications to the parameter change card (PARMCHNG) are discussed first. Changes to the other descriptors and to the program control card and user interface are then discussed. Section 3 is a discussion of the software documentation and introduction to the individual parts of the program documentation. Because of the large volume of documentation, some material is included as appendices rather than within Section 3.

Two of the appendices need special note. This technical report is written with the assumption that the reader is familiar with BALFRAM, especially the use and functioning of the BALFRAM input cards (called descriptors). For readers unfamiliar with the descriptors, Appendix D provides a brief overview of the purpose of each descriptor.

Dr. Paul Tuan presented a paper at the Theater-Level Gaming Conference in Washington, D.C. in September 1977. His paper, entitled "Some Tactical Problems in Man/Computer Interactive Gaming," centered around BALFRAM, and is included with the accompanying illustrations in this report as Appendix E.

2. BALFRAM PROGRAM REVISIONS

2.1 INTRODUCTION

BALFRAM has been used for military studies and planning since the early 1970s by several major military commands. During this time, analysts have defined revisions that would make BALFRAM easier to apply to typical military problems or that would enhance the user interface through revised inputs and outputs. The task work reported here is the first task in a logical progression leading to an interactive BALFRAM. These changes represent a beginning of the updating process.

Because BALFRAM is a large, complex model being used in several commands, the change process proceeded in a conservative, methodical way. Because the changed program would be converted to other computers, ANSI standard FORTRAN was followed. FORTRAN statements unique to a particular computer were avoided as an impediment to transferability. When converting the original BALFRAM to the SRI CDC 6400 computer, the changes were made in a form that would require little or no conversion effort when transferring back to the Honeywell WWMCCS computer. The software improvements also were implemented in standard FORTRAN with liberal use of comments.

Where new capabilities were added to old BALFRAM descriptors or where new descriptors were added, existing BALFRAM input formats were followed as closely as possible. All the original BALFRAM descriptor cards continue to function as described in BALFRAM User Manual so that existing data decks can be used as they are, with the improved program. BALFRAM users can incorporate the improvements without a major training process or disruptions of existing uses of the model. With this approach, there was no need to make major changes to the input routines--a job that would have to be redone during the design and implementation of the interactive version of BALFRAM.

Software changes were designed to minimize the memory and execution time required. New functions were added to existing descriptors rather than defining new descriptors and new variables. Changes were made to allow as much processing as possible outside the innermost loop of subroutine TSMO, where event and time updates occur during the progression of simulated time.

Design and program coding of any changes were done in a straightforward manner. Unusual methods that might save minor amounts of memory or execution time (such as word packing or complex FORTRAN programming) were avoided in the interest of program comprehension and maintainability.

A series of subtasks were specified in the SRI proposal and contract. Additional changes were made as they were required. For clarity of discussion, these changes can be organized into the following areas:

- The PARMCHNG descriptor
- Other descriptors
- Program control and user interface.

2.2 PARMCHNG DESCRIPTOR IMPROVEMENTS

2.2.1 Background Leading to Improvements

The PARMCHNG (parameter change) descriptor resets input parameter values at a specified time during a BALFRAM scenario. This descriptor card specifies the unit and parameter that is to be changed and provides the new parameter value and the time this new value is to take effect. Several changes were needed to improve the utility of this card.

The original PARMCHNG card specified only one unit, one parameter, and one effective time. However, typical scenarios might require the same parameter be changed for several units at the same time or several parameters changed for one unit at once. These situations required a multiple of almost identical PARMCHNG cards, making the input cumbersome to assemble.

During sensitivity studies, RANDMSEQ and FRCRATIO cards often vary the same parameters that are changed by PARMCHNG cards. The RANDMSEQ

and FRCRATIO cards affect the initial game conditions according to the input instructions. The PARMCHNG card replaces parameter values during the play of the game. In choosing PARMCHNG values, the analyst usually relates the new value to the initial value of the parameter so that the new value is proportional to the initial value. Although the RANDMSEQ and FRCRATIO cards would scale the initial values, they would not scale the new values on the PARMCHNG card, thereby destroying the proportion between the initial conditions and the updated conditions. This caused inaccuracies in the sensitivity runs, reducing the value of automatic generation of sensitivity studies.

The variables to be changed by the PARMCHNG cards are identified by a code that can specify nine variables, primarily variables on the UNITSPEC card. Use of the BALFRAM model in typical analytical situations indicated that additional variables should be able to be changed. These included variables on the NODEPROP card, which cannot be changed at all in the original mode. These additions would provide more flexibility in describing the complexities of military situations.

2.2.2 Change to Permit Multiple Unit References on a PARMCHNG Card

To accommodate multiple changes on one PARMCHNG card, three new descriptor cards were implemented. These new cards describe three types of multiple change situations. Table 1 compares the number of units, parameter codes, and parameter values per card of the original PARMCHNG descriptor cards with the capabilities of the three added descriptor cards: PVCHANGE, UVCHANGE, and UNCHANGE. The PVCHANGE descriptor card permits changing up to four parameters for a specific unit. The UVCHANGE descriptor card permits changing the same variable code for up to four different units by providing a new parameter value for each of the four units. The UNCHANGE descriptor card is similar to the UVCHANGE descriptor except that a single parameter is changed to a given value, but it can be changed for up to ten units with the one input card. The descriptor names refer to the fields changed. PVCHANGE changes multiple Parameter codes and Values; UVCHANGE changes multiple Units and Values; and UNCHANGE changes multiple UNits.

Table 1

CAPABILITIES OF NEW PARMCHNG DESCRIPTORS
(Table entry is the number of descriptor fields entered on one card)

Descriptor Name	Descriptor Field			
	Effective Time	Unit Identification	Parameter Code	New Parameter Value
PARMCHNG	Single	Single	Single	Single
PVCHANGE	Single	Single	4 Maximum	4 Maximum
UVCHANGE	Single	4 Maximum	Single	4 Maximum
UNCHANGE	Single	10 Maximum	Single	Single

Depending on the scenario updates to be made, each new descriptor can save from three to nine cards in the input files. For example, if 10 aircraft units (squadrons, for example) were to be augmented with four aircraft each 30 days after the start of the scenario, a single UNCHANGE card could be used to describe the situation instead of the ten PARMCHNG cards previously required. If the squadrons, however, were to be augmented with differing number of aircraft per squadron, then the UVCHANGE card would be used where the unit identifier for each squadron and the number of aircraft augmented for that squadron would be entered for up to four squadrons per card.

The FORTRAN program changes that implement the new descriptors effect only the input processing. During the input edit processing, the multiple values on the new descriptors are expanded to a series of equivalent single PARMCHNG cards. The reordering by time of parameter changes in subroutine MINV and the updating of the PARMCHNG values during the play of the battle scenarios in subroutine ADCP5 remain the same. The maximum limit of 100

parameter changes is also retained. Software changes are only necessary in subroutine INPUTB and do not add memory or processing requirements to the critical overlays in the actual battle processing.

The new card formats and data formatting instructions are included in Appendix A. These instructions should be added to the BALFRAM User Manual after the discussion of the PARMCHNG descriptor. The card formats have been designed to adhere to the BALFRAM standard of five-and-ten-column field definitions (although unit identifiers are generally three columns wide).

2.2.3 Additional New PARMCHNG Variable Codes

To extend the PARMCHNG card capability, four new variable codes were added to permit changing parameters found on the NODEPROP card. All fields on the NODEPROP descriptor were examined and only four fields were found appropriate to change by means of the PARMCHNG descriptor. The major NODEPROP fields not changed are the exogeneous firepower and time values. The NODEPROP exogeneous firepower effectiveness and its associated fields (scaling factor and fraction of order of battle to generate exogeneous firepower) can already be changed by code 2 on the PARMCHNG descriptor card. The NODEPROP time values were not included because of the interacting effects of multiple time changes and the possibilities of difficult contingency logic debugging when time changes occur in more than one place.

The new values of the parameter change codes are shown in Table 2. The first nine codes are the same as the existing codes; the last four (numbers 10 through 13) are the newly added NODPEPROP variable codes. Table 2 provides the new code values as well as referencing the descriptor and field location where the complete variable definition can be found in the User Manual. These new codes are to be used on the PARMCHNG, PVCHANGE, UVCHANGE, and the UNCHANGE descriptors.

The new variable codes were implemented in subroutine ADCP5, where the PARMCHNG processing is accomplished during the scenario play. New FORTRAN statements were added for processing codes 9 through 13 and a

Table 2

REVISED PARMCHNG VARIABLE CODES

Variable Change Code	Variable Definition	Variable Definition Reference	
		Descriptor Card	Field Number
1	Increment to unit order of battle	UNITSPEC	5
2	Not used		
3	Index of Combat Effectiveness (ICE)	UNITSPEC	8
4	Mobility factor	UNITSPEC	10
5	Nodal objective of unit	UNITSPEC	12
6	Not used		
7	Exogenous firepower target node	UNITSPEC	13
8	Disengagement code	GUERILLA	5,7,9,...
9	Defeat criterion (actual order of battle - not percent)	UNITSPEC	7
10	Nodal location of units used to compute exogenous firepower	NODEPROP	4
11	Target node for exogenous firepower	NODEPROP	5
12	An additional node whose use is determined by the code supplied in field 7 - NODEPROP	NODEPROP	6
13	Exogenous firepower computation code	NODEPROP	7

branch was inserted into the existing code to transfer to the new section if the code value was greater than 9. This change permitted implementation without substantial reprogramming of subroutine ADCP5.

2.2.4 PARMCHNG Adjustment to Variables Affected by FRCRATIO and RANDMSEQ Descriptors

To properly scale PARMCHNG variables for sensitivity studies, new FORTRAN programming was inserted to save the values input on the PARMCHNG cards for subsequent scaling by FRCRATIO and RANDMSEQ processing. Since initial conditions are already saved twice (after input has been read and edited and after FRCRATIO and DSTRIBUT variations are applied to the data), the PARMCHNG data could be saved on the same files. Thus, the initial PARMCHNG conditions could be retained the same as the other program data even after scaling by the FRCRATIO and RANDMSEQ parameters.

Scaling PARMCHNG values is required during FRCRATIO processing in subroutine SURFGN and during RANDMSEQ processing in subroutine RANDM. In both places, PARMCHNG scaling is required only when identical variables from the identical units matched. Although identical variables are changed by PARMCHNG, FRCRATIO, and RANDMSEQ descriptors, each descriptor used a different variable code to identify variables. To efficiently match identical variables, two translation tables were constructed for converting the FRCRATIO and RANDMDEQ variable codes to the equivalent PARMCHNG code. These tables (in arrays IFR2PC and IRS2PC) were built into subroutines SURFGN (for FRCRATIO) and subroutine RANDM (for RANDMSEQ).

The general procedure for PARMCHNG scaling is the same for both FRCRATIO and RANDMSEQ processing. After computing FRCRATIO and RANDMSEQ scaling factors, all PARMCHNG cards are checked for a match on unit identifier and PARMCHNG variable code. When this match occurs, the scaling factor is applied to the PARMCHNG new value. The scaling factors are then applied to the variables specified on the FRCTATIO or RANDMSEQ cards. As BALFRAM loops through variations in sensitivity studies, PARMCHNG variables are automatically reset to initial conditions with the other program data by being read from the appropriate files. This method requires additional processing during the initial phase of each sensitivity run rather than

in the game simulation loop during the play of a particular variation. Such placement saves execution time by minimizing calculations in the innermost loops of the game-play subroutines.

While subroutine RANDMN was being improved with the new PARMCHNG programming, the format of the summary data that are printed was also changed to print more information and to label the printed information more descriptively.

2.3 CHANGES TO OTHER DESCRIPTORS

In addition to the major changes in the PARMCHNG descriptor, other improvements increased the capabilities of or added functions to other BALFRAM descriptors. These improvements expanded BALFRAM capability for describing typical military situations. This section describes these changes.

2.3.1 UNITSPEC Descriptor Sortie Rate

Some units in a military scenario have attrition expressed on a per sortie basis. An example is an aircraft squadron where attrition is measured on a sortie basis--one aircraft performing one mission. The individual equipment might perform one mission more or less than once per day depending on the scenario and equipment capability. Aircraft could fly 1.5 sorties per day on the average (3 sorties every 2 days) or 0.5 sorties per day (1 sortie every other day). In assembling data for equipment with sortie rates different than 1.0, the analyst would have to multiply the equipment inventory by the sortie rate for input as the order of battle. Handling data this way, however, obscures the quantity of equipment available and the sortie rate.

BALFRAM was changed to make the sortie rate an explicit input on the UNITSPEC card in place of a variable no longer used. During input editing in subroutine INPUT, sortie rate (i.e., missions per time unit) and order of battle (now interpreted as on-hand equipment) are multiplied to give the order of battle for simulation play. When sortie rate is not applicable, 1.0 is entered so the order of battle remains as specified on the UNITSPEC card.

2.3.2 Increased Units on LGINTDIC and SUMUNIT Descriptors

As experience was gained with BALFRAM, it was found that the number of units specified on the LGINTDIC and SUMUNIT descriptors were inadequate to describe scenarios modelled. This situation was remedied by increasing the size of the arrays for storing the unit identifiers. In addition, parameters controlling the input editing of these descriptors were adjusted to indicate the increase of units from 10 to 20. The input READ statement also had to be adjusted to allow a second data card if required. No changes were required in subroutines ADCP4 or ADCP7 where the LGINTDIC and SUMUNIT parameters are processed because this processing is controlled by indexes constructed during input editing.

2.3.3 New STOPBTLE Descriptor Functions

The only method to stop the simulation play in the original BALFRAM model was based on the destruction of all units on an input list. Extensive use of BALFRAM indicated that other criteria based on time and unit objectives would be useful for stopping the battles. Thus, two more criteria have been added--stopping when the duration of the battle reaches a predefined time and stopping when any one of a list of units reaches its final objective.

Rather than introduce new descriptor cards, the STOPBTLE card was modified to accommodate these two new functions along with the function it already performs. The first unit in the list of units on the STOPBTLE card is now used as a code value. If this unit is positive, the STOPBTLE card processes as it always has. If the field for the first unit contains -1, the field for the second unit is interpreted as the time to end the battle. If the field for the first unit contains -2, the battle is stopped if any unit in the list of units has reached its final objective. New programming was added to subroutine ADCP to process the new forms of STOPBTLE and to subroutine SHFREF to eliminate the internal unit identifier conversion.

Appendix A contains descriptions of the new functions of the STOPBTLE card for insertion into the appropriate place in the BALFRAM User Manual.

2.3.4 Added PROASIGN Descriptor Function

The original PROASIGN descriptor redistributed the total surviving order of battle of a set of units back among those units according to input apportionment factors. These factors could be changed as a function of time. As complex scenarios were developed, a new requirement was evident. The order of battle to be apportioned could be a function of the remaining order of battle of another unit upon whom the list of units were dependent. Such a case might be the air wings on a carrier. As the carrier is damaged, two effects occur: aircraft on board the carrier are also damaged, and the capacity of the carrier to launch the recover aircraft is diminished so that all the undamaged aircraft on board cannot be launched. To account for these effects, a second function was added to the PROASIGN descriptor. This work was not done under this contract, but it is reported here to update the BALFRAM documentation.

The new PROASIGN function computes the sum of the remaining order of battle of a set of units and compares it with the product of a given constant multiplied by the remaining order of battle of a specified unit. The minimum value of the comparison is then allocated to the set of units according to the apportionment factors.

Appendix A contains a description of the new PROASIGN function for insertion into the appropriate place in the BALFRAM User Manual.

2.3.5 Program Control Card Changes

Several minor changes were made on the Program Control Card to permit input of additional control information.

During this contract, fields 17 to 26 (columns 51 to 70) were added as input flags to control BALFRAM processing. Only fields 17 and 18 currently have significance--the remaining fields are available for future use. Field 17 controls the production of a file for use with a plotting program. SRI wrote a small, off-line plotting routine to test the capability for plotting, but no plotting capability was included in BALFRAM at this time. Field 18 adds another control to the printing of the battle history. When field 18 contains a 1 battle history is printed only if

an event other than a time update occurs. This feature limits the output to only significant events.

Other changes were made to the Program Control Card prior to the current SRI contract, but they were not formally documented previously. Field 5 was originally used only to control the number of battle steps before beginning the battle history printout. As an added feature, the battle history can be selectively printed every nth step by entering -n in field 5. A +n works as originally specified. A new field 6 (columns 26-30) has been added that controls the reading of the NODH geography. To read NODH data from file generated by the NODH program a zero or blank is entered in field 6. New fields 7-16 (columns 31-50) have been reserved for a user supplied program logic indicator array. Although read and printed this array is not otherwise used apparently as a result of other program modifications.

The current version of the Program Control Card is documented in Appendix A.

2.4 CHANGES IN PROGRAM CONTROL AND USER INTERFACE

In addition to the changes in the functioning of certain descriptors, changes were made in the subroutines that control the BALFRAM program and that read and print data. The program was streamlined and the output formats changed to make the presentation of data easier for the user to interpret.

2.4.1 Restructure the BALFRAM Software

The BALFRAM program used as a starting point for this research contained 20 subroutines used for computing allocation of forces in a dynamic situation (called the SABRE GRAND model). Research subsequent to the introduction of these subroutines into BALFRAM showed that the theoretical basis of these routines was not compatible with BALFRAM. These routines were never used in BALFRAM scenarios, but had not been removed from the program. These subroutines and the storage associated with them were removed.

Overlays (or links in Honeywell terminology) are a method of segmenting a program so that only portions that are needed for immediate execution are in memory. This permits large programs to execute in a smaller memory space. The original BALFRAM overlay structure used overlay methods unique to Honeywell computers. These methods increased the effort necessary to transfer BALFRAM to other commands or other potential users.

A new overlay structure was designed and implemented that provided a tree-like hierarchy of subroutine calling sequences and overlay partitions. This sequence can be implemented on the Honeywell as well as other computers (such as the CDC 6400 at SRI) with only minor syntax changes to reflect the host computer linkage syntax.

Figure 1 shows the new overlay structure and subroutine calling sequence. The new structure requires copies of some subroutines (LLRK1, MLRK1, and the geometric subroutines) in two overlays. This is implemented easily by placing these subroutines in a user library so that the loader has access to the routines for loading both overlays, yet only one copy of the FORTRAN programs need be maintained.

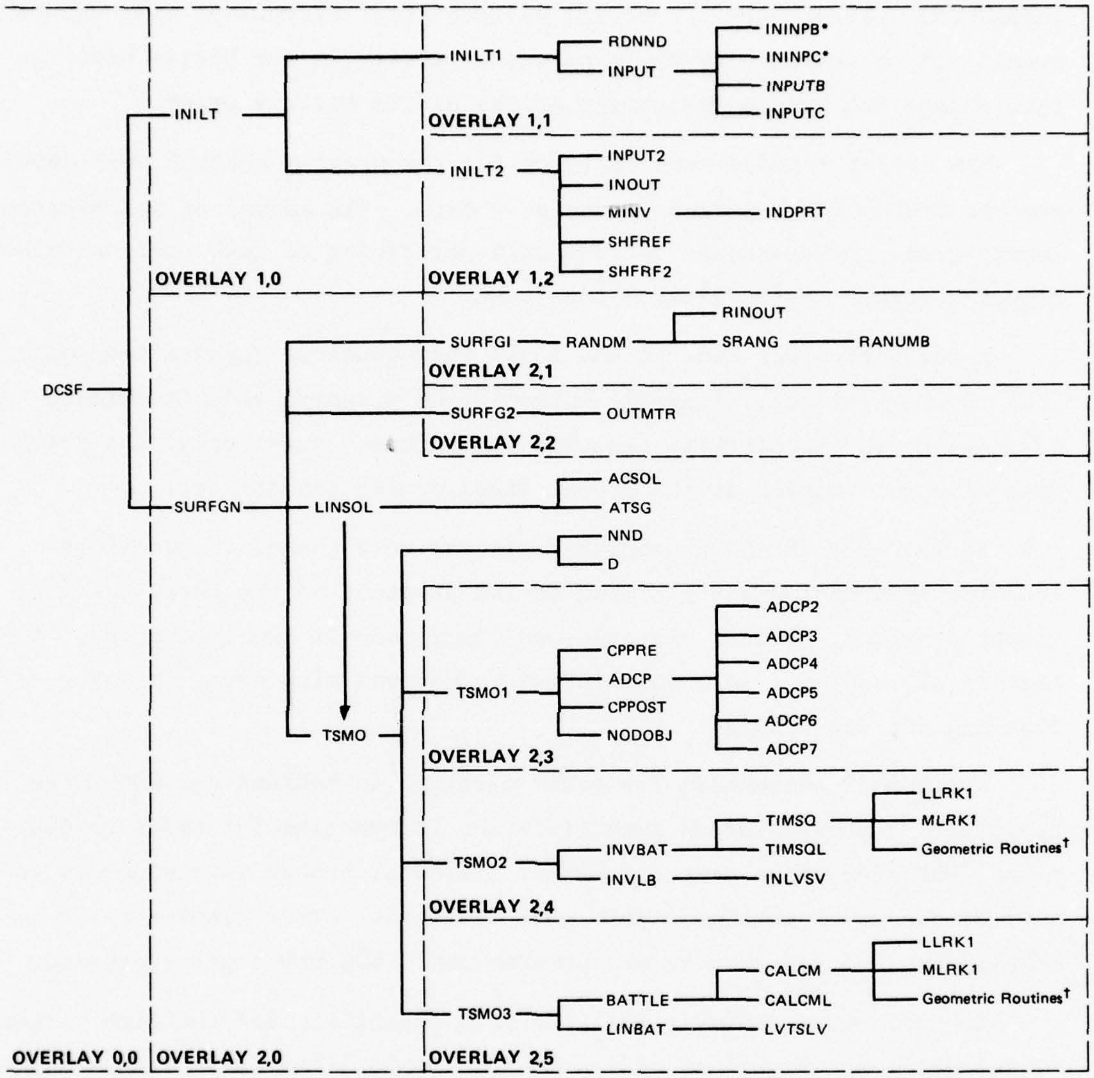
2.4.2 Corrections to the Nonhomogeneous Linear Battle

One of the BALFRAM program difficulties identified by analysts was the inability to run nonhomogeneous linear battles with surveillance. Analyst intuition and manual checking revealed that the answers produced were unreasonable. Further analysis and test cases run by SRI eventually traced the error to a typographical error in a variable name in subroutine INVLB, which subsequently was corrected.

2.4.3 Revision of BALFRAM Outputs

Several changes were made to make BALFRAM outputs easier to read by reformatting the output or correcting errors in producing the outputs.

If the full battle history printout is selected for a BALFRAM scenario, the resultant output is an extremely large amount of paper, only part of which is useful. The large quantity of paper is caused by the



*Entry Point

†*Geometric Routines include:*

ACOSH	EXM1
ACOTH	EXM2*
ACSSCH	EXM3*
ASEC	HCSCT
ASINH	SECTAN
COSH	SINH

SA-5822-1

FIGURE 1 BALFRAM SUBROUTINE LINKAGE SEQUENCE

number of events triggered by both battle contingencies and by the passage of simulated time. One of the new option indicators that can be set at input time will disable the battle history print of routine time update events, which leaves only the contingency events in the battle history. This change can reduce the amount of the battle history print.

New output formats were designed for the program control card data and the BTLENODE-Homogeneous descriptor data. The amount of descriptive labeling was increased and all the data pertaining to each node was displayed together to facilitate comparison.

A correction was made to the logic that controls the printing of FEBA movement history, logistic interdiction history, and interpreted descriptors for sensitivity analysis iterations. Previously, the printouts were not enabled at the proper times during the battles.

Formats for the NODH geography program were changed. Additional labeling information was included on the printout of the three matrices (input distance, minimum distance, and next node in shortest path). A listing of the input in a descriptive format was also added for ease of checking the input data.

All FORMAT statements for error messages in BALFRAM and NODH were rewritten to provide added identification information for error resolution. The subroutine name and FORMAT statement number were added to the text of the error message. This will facilitate error resolution by identifying the location in the program where the error was processed.

To facilitate adding a full plotting capability for the time variation of selected variables, one of the BALFRAM input files (file 1) was converted to a plot file. This file was originally used only during the input processing and was therefore able to be additionally used for plotting data during the simulation. An option switch was established in the input to enable writing the plot tape when desired. A small off-line plotting routine was written for the SRI version of BALFRAM to test the feasibility of plotting data.

2.4.4 Revisions to Input Processing

Two changes were made to the processing of input data. The original version of BALFRAM printed all the input data twice--once with the input cards numbered consecutively as they were read into the program and a second time when each card was edited. The second list was intended to be used when there were editing errors. This second list was changed to print input only when an error is present.

Each section of the input processing that reads a descriptor with the designation "Red" or "Blue" contains program statements to interpret the color. Since 26 of 33 descriptors have such information, there is much redundant programming. In preparation for the major design changes of interactive BALFRAM, a code was written to interpret the side only once during the input processing. New descriptors added during this contract and any descriptor input processing that is changed will take advantage of this coding to reduce the program length for processing input.

To improve the flexibility of BALFRAM for future software changes, a new COMMON storage area has been inserted. This COMMON has a floating point and a fixed point array that can be used when communication is required between subroutines for debugging or testing. This COMMON also includes an array for selecting various processing options. Currently, only two of the options are utilized although there is space for up to 20 options.

Preceding Page Blank - Not Filmed

3. SOFTWARE DOCUMENTATION

3.1 INTRODUCTION

As part of this research on an interactive BALFRAM program, SRI updated the software documentation of the BALFRAM program. In addition, to updating existing documentation to indicate the current status of the model after the most recent changes, new documentation was developed. Section 3 discusses this documentation. The purpose of this documentation work is to make the design and implementation of interactive BALFRAM faster and more efficient by providing tools for identifying and understanding the pertinent sections of the BALFRAM software.

Some documentation is too voluminous to be included in this section of the report and is more appropriately placed in appendices. The appendix material, however, is discussed in this section. Information discussed previously in the report is also referenced in this section to provide in one section a complete discussion of software documentation.

3.2 BALFRAM PROGRAM STRUCTURE

In subsection 2.4.1, the restructuring of the BALFRAM software was discussed. In that section, Figure 1 showed the overlay structure and subroutine calling sequence. Figure 1 is also important software documentation for understanding the physical organization of the BALFRAM program and the hierarchy of references between subroutines. Further information on the purpose of each subroutine can be found in the BALFRAM Program Maintenance Manual, Section 2.2.2.1. Additional information on the logic and local variables of each subroutine can be found in the BALFRAM Program Notebook developed during this research and included with the BALFRAM working notes in the Research and Analysis Office (J77) at CINCPAC.

3.3 BLOCK DIAGRAMS OF BALFRAM CONTROL ROUTINES

To document more fully the logical working of the BALFRAM program, the major control subroutines were block diagrammed during this research. The block diagrams show the flow of program processing at a macro-level of detail. Individual FORTRAN statements are not shown, but collections of statements that perform particular functions are shown along with the logical branches that control the execution of the major program sections. At this level of detail, the important program construction and logical features are evident without the confusing burden of detailed variable processing. The block diagrams are keyed to the FORTRAN program by referencing program statement numbers so that specific details in the program can be readily identified. Table 3 provides a list of those subroutines for which block diagrams have been drawn. Because of the volume of the block diagrams, they are included in Appendix B.

3.4 COMMON VARIABLE DEFINITIONS

The information content of variables is critical to understanding and changing a program. A dictionary of the variables in each COMMON block of BALFRAM has been assembled and updated during this research. Because of the number of variables, the tables containing the definitions are presented in Appendix C.

In the BALFRAM program, variables are assembled into COMMON blocks according to their usage. Generally variables associated with a particular descriptor are organized into the same COMMON block. To preserve the correspondence of variables and their COMMON blocks, the variable definitions are also organized by COMMON block. All variables in each COMMON block are defined, their size (DIMENSION) given, and the source of the data is provided for cross reference.

3.5 SUBROUTINE COMMON REFERENCES

When interpreting and debugging programs it is important to know the information contained by each variable and the subroutines where the variable values are defined and referenced. The information definition

Table 3

SUBROUTINE BLOCK DIAGRAM LIST

Subroutine Block-Diagrammed	Subroutine Purpose
INILT	Reads and edits input descriptors for errors.
SURFGN	The main driver for battle computations. Establishes initial conditions for battle via FRCRATIO, DSTRIBUT, and RANDMSEQ control instructions.
RANDM	Randomizes designated input parameters according to RANDMSEQ card set inputs and assigns variable values for individual battles.
TSMO	The main battle scenario driver. Controls scenario by determining next event and time. Writes battle history.
BATTLE	Calculates attrition during the interaction of homogeneous forces.
INVBAT	Computes the time required for units engaged in homogeneous battle to reach their respective defeat criteria.
LINBAT	Calculates attrition during linear, nonhomogeneous interactions.
INVLB	Computes the time required for units engaged in linear nonhomogeneous battle to reach their respective defeat criteria.
ADCP5	Processes parameter changes as required by the PARMCHNG descriptors.

was discussed in Section 3.4. The subroutine usage of variable data is discussed in this section.

Compiler variable maps give only definition and reference information relative to a single subroutine (intra-subroutine). For tracing, variable references are needed on a global basis (inter-subroutine). Special programs must be executed to obtain such information. In the absence of complete global information, knowledge of the COMMON references by subroutine are useful in tracing the usage of a particular variable. Table 4 provides the map of COMMON block references by subroutine for BALFRAM. In Table 4, individual subroutines can be identified as users of a COMMON containing a variable of interest. Reference to the compiler variable map can then be used to identify the usage of the particular variable.

3.6 BALFRAM FILE DESCRIPTIONS

The BALFRAM program uses disk files to read input, to write output for the printer, and to store intermediate results for minimizing the requirements for memory. Table 5 describes the files used by BALFRAM. The File Logical Unit Number is the unit number referenced by the READ or WRITE statements in the program. On the WWMCCS computer, this number is linked to a physical file through a FILE or PRMFL control card in the job control deck depending on the disposition of the file. On the SRI CDC 6400 computer, the file logical unit number is linked to a physical file through the file designations on the FORTRAN PROGRAM card and through the job control cards.

Table 5 indicates the subroutines where files are used and the information content of the files. Files 1 and 2 are created by copying the input file and are used to read each input card twice. The first read (from file 2) is used to determine the type of descriptor of the input card. After branching to the appropriate section in subroutine INPUT, the descriptor card is read from file 1 according to the proper FORMAT for that descriptor.

Files 50, 52, and 54 store variable initializations when sensitivity studies are to be generated. FRCRATIO, DSTRIBUT, and RANDMSEQ descriptors

Table 4

COMMON BLOCK REFERENCES--USAGE BY SUBROUTINE

Overlay		0,0		1,0	1,1					1,2						
COMMON Name		Subroutine Name														
Name	Size	DSCF	EXIT	INILT	INILT1	RDNND	INPUT	INPUTB	INPUTC	INILT2	INPUT2	MINV	SHFREF	SHFRF2	INOUT	INDPRT
BLUE	2520	X		X			X		X		X	X			X	
RED	2520	X		X			X		X		X	X			X	
DCMN	2712	X		X		X	X		X		X	X	X	X	X	
F	217	X					X				X	X	X		X	
H	301	X					X				X	X	X		X	
E	8636	X					X	X	X		X	X	X		X	
G	3704	X					X	X	X		X	X	X		X	
SUMRY	2553	X					X				X	X	X		X	
CALCMC	554	X		X			X				X	X			X	
L	4613	X					X				X	X	X		X	
NLOG	10	X		X							X					
PATCH	8	X		X							X	X				
LSOLA	25	X						X					X		X	
R	683	X						X					X		X	
LG	766	X						X					X		X	
FB	765	X						X				X		X	X	
SUNIT	231	X						X						X	X	
FBA	15	X														
PLN	1080	X														
PLAN56	10	X														
TEMP	130	X		X												
OL11	3	X			X					X						
OVL22	22	X														
OVL23	1	X														
OVL245	3	X														

Table 4 (Continued)

Overlay		2,0							2,1					2,2	
COMMON Name		Subroutine Name													
Name	Size	SURFGN	LINSOL	ACSOL	ATSG	TSMO	D	NND	SURFG1	RANDM	SRANG	RINOUT	RANUMB	SURFG2	OUTMTR
BLUE	2520	X	X			X				X		X			X
RED	2520	X	X			X				X		X			X
DCMN	2712	X	X			X	X	X		X		X			X
F	217	X										X			
H	301	X										X			
E	8636	X								X		X			
G	3704											X			
SUMRY	2553		X									X			X
CALCMC	554					X						X			
L	4613					X						X			
NLOG	10														
PATCH	8	X				X									
LSOLA	25	X	X									X			
R	683	X								X	X	X			X
LG	766											X			
FB	765											X			
SUNIT	231											X			
FBA	15					X									X
PLN	1080														
PLAN56	10														
TEMP	130					X									
OL11	3														
OVL22	22		X											X	
OVL23	1					X									
OVL245	3					X									

Table 4 (Continued)

Overlay		2,3										
COMMON Name		Subroutine Name										
Name	Size	TSMO1	ADCP	ADCP2	ADCP3	ADCP4	ADCP6	ADCP5	ADCP7	CPPRE	CPPOST	NOBOBJ
BLUE	2520		X	X	X	X	X	X	X			X
RED	2520		X	X	X	X	X	X	X			X
DCMN	2712		X	X	X	X	X	X				X
F	217		X	X	X							
H	301		X	X	X							
E	8636		X	X	X	X						
G	3704		X	X	X							
SUMRY	2553											
CALCMC	554		X	X	X		X					
L	4613		X									
NLOG	10											
PATCH	8											
LSOLA	25											
R	683											
LG	766					X						
FB	765							X				
SUNIT	231								X			
FBA	15						X					
PLN	1080		X	X	X	X						
PLAN56	10						X	X				
TEMP	130											
OL11	3											
OVL22	22											
OVL23	1	X										
OVL245	3											

Table 4 (Continued)

Overlay		2,4								
COMMON Name		Subroutine Name								
Name	Size	TSMO2	INVBAT	TIMSQ	TIMSQL	INVLB	INLVSV	Geometric Subroutines*	MLRK1	LLRK1
BLUE	2520		X			X				
RED	2520		X			X				
DCMN	2712		X	X						
F	217									
H	301									
E	8636									
G	3704									
SUMRY	2553									
CALCMC	554	X	X							
L	4613	X				X	X			
NLOG	10									
PATCH	8									
LSOLA	25									
R	683									
LG	766									
FB	765									
SUNIT	231									
FBA	15									
PLN	1080									
PLAN56	10									
TEMP	130									
OL11	3									
OVL22	22									
OVL23	1									
OVL245	3	X								

*Geometric Subroutines include:

ACOSH	ASEC	EXM1	SINH
ACOTH	ASINH	HCSCT	
ACSSCH	COSH	SECTAN	

Table 4 (Concluded)

Overlay		2,5								
COMMON Name		Subroutine Name								
Name	Size	TSMO3	BATTLE	CALCM	CALCML	LINBAT	LVTSLV	Geometric Subroutines*	MLRK1	LLRK1
BLUE	2520		X			X				
RED	2520		X			X				
DCMN	2712		X				X			
F	217									
H	301									
E	8636									
G	3704									
SUMRY	2553									
CALCMC	554	X	X							
L	4613	X				X	X			
NLOG	10									
PATCH	8									
LSOLA	25									
R	683									
LG	766									
FB	765									
SUNIT	231									
FBA	15									
PLN	1080									
PLAN56	10									
TEMP	130									
OL11	3									
OVL22	22									
OVL23	1									
OVL245	3	X								

*Geometric Subroutines include:

ACOSH	ASEC	EXMI	SINH
ACOTH	ASINH	HCSCT	
ACSSCH	COSH	SECTAN	

48

Table 5

BALFRAM FILE DESCRIPTIONS

File Logical Unit Number	Mode	Written in Subroutine	Read in Subroutine	Contents
1	BCD	INPUT, TSMO	INPUT, INPUTB INPUTC	Input data transferred from file 5: used to read each descriptor according to the proper FORMAT; subsequently used for data to be plotted after input processing is completed.
2	BCD	INPUT	INPUT, INPUTB INPUTC	Input data transferred from file 5: used to determine descriptor type of input.
5	BCD	--	INPUT	Input data--standard input file.
6	BCD	many	--	Printer output data--standard output file.
20	Binary	NODH program	RDNND	Geography data from program NODH--node-to-node distance matrix (DISTPK), node-to-node distance matrix indexes (NODEPK), and next-node-in-path matrix (NEXTPK).
42	Binary	OUTMTR	OUTMTR	Each record contains the battle results of all SUMMARY cards for a given set of FRCRATIO, DSTRIBUT, and RANDMSEQ variations.
50	Binary	SURFGN	SURFGN	All descriptive variables for each side (COMMON BLUE and RED) and the PARMCHNG value array as initialized by the input data.
52	Binary	LINSOL	LINSOL	Same contents as file 50 stored after a set of FRCRATIO and DSTRIBUT variations have been applied to the variables.
54	Binary	SURFGN	SURFGN	Same as file 52.

scale variables with multiplicative constants. For each variation, the program variables must be restored to the appropriate initial values by reading one of the files. File 42 stores the summary values for each variation in the sensitivity studies to minimize the memory required. After the last sensitivity variation, file 42 is read, and the summary output formatted and printed by subroutine OUTMTR.

3.7 DESCRIPTOR PROCESSING

In designing and implementing interactive the BALFRAM program, descriptor processing will be altered. To make changes, it is important to know where the processing for each descriptor takes place. Users of the existing BALFRAM also need such a cross reference to make their own changes to descriptor processing. Table 6 provides a reference to the subroutine where each descriptor is processed, both during input editing and during the execution of the contingency logic. Since all input and editing is done in three very large subroutines, statement numbers are furnished to further locate the input processing sections.

Table 6

BALFRAM DESCRIPTORS PROCESSING CROSS REFERENCE

Descriptor Acronym	Location of Input Processing		Location of Logical Processing	Descriptor Purpose
	Subroutine Name	Statement Number	Subroutine Name	
Program Control Instructions				
Title Card	INILT	12	None	Descriptive title information
Program Control Card	INILT	20	INILT	Sets general program parameters
LINSOLU	INPUTB	2000	LINSOL	Generates the order of battle required to produce a stipulated outcome
SUMMARY	INPUT	1000	OUTMTR	Summarizes survivors, firepower results, duration of battle
SUMTITLE	INPUT	1900	OUTMTR	Provides headings for summaries
END	INPUT	9998	INPUT	Signals completion of input
Force Definition Inputs				
UNITSPEC	INPUT	100	None	Describes combat forces
GUERILLA	INPUT	1800	INPUT	Establishes disengagement rules
PARMCHNG*	INPUTB	3000	ADCP5	Changes parameters, such as order of battle, firepower
Battle Logic Inputs				
BTLENODE	INPUT	1100	ADCP	Describes nodes for homogeneous and non-homogeneous battles
STOPBTLE	INPUTC	700	ADCP	Terminates the battle
NODATFAC	INPUT	1300	ADCP3	Sets attrition factors for specified conditions
NODEPROP	INPUT	500	ADCP	Specifies allocation and effectiveness of supporting firepower
SUMUNIT	INPUTB	3100	ADCP7	Permits orders of battle of several units to be merged
EXOGUNIT	INPUTB	2300	ADCP	Directs firepower to enemy units at nodes occupied by friendly units
OPEXOGUN	INPUTB	2400	ADCP	Directs supporting firepower at the locations of specified units
PROASIGN	INPUT	1400	ADCP2	Proportionally assigns and redistributes forces
LGINTDIC	INPUTB	2600	ADCP4	Describes logistic pipelines and effects of interdiction
Movement Logic Inputs				
ADVANCE	INPUT	400	ADCP2	Moves units from node to node contingent on arrival events
RETREAT	INPUT	600	ADCP2	Moves units contingent on defeat events at nodes
OBJCTADV	INPUT	1200	ADCP3	Relocates units when enemy is defeated
STRTEGRT	INPUT	1700	ADCP3	Withdraws units if force ratio is unfavorable
TIMEADVN	INPUT	2200	ADCP	Relocates units at specified times
CHASE	INPUTC	1500	ADCP2	Causes one force to track another
RENDEVOU	INPUTC	1600	ADCP2	Establishes a sequential link-up of forces
REDEPLOY	INPUTC	300	ADCP	Redeploys units after battle is won
FEBAMOVE	INPUTB	2900	ADCP6	Traces movement of forward edge of the battle area
Sensitivity Analysis Instructions				
FRCRATIO	INPUT	200	SURFGN	Multiplies order of battle, ICE, or firepower to vary scenario outcome
DSTRIBUT	INPUT	900	SURFGN	Allocates forces between two missions and/or areas
RANDMSEQ	INPUTB	2500	RANDM	Generates parameter sensitivity studies

*Also: PVCHANGE, UVCHANGE, UNCHANGE.

Appendix A

REVISED USER MANUAL DESCRIPTOR DOCUMENTATION

Preceding Page BLANK - NOT FILMED

Appendix A

REVISED USER MANUAL DESCRIPTOR DOCUMENTATION

1. INTRODUCTION

During the interactive BALFRAM research, new functions were added to several of the descriptor cards. Because these modifications affected the input of data, the descriptions of input data assembly in the User Manual* were outdated. The following sections of this appendix update the descriptor documentation to include new functions and capabilities. This new documentation should replace existing documentation in the User Manual. New documentation is included for:

- Program control card
- Parameter change cards
- STOPBTLE card
- PROASIGN card
- UNITSPEC, LGINTDIC, and SUMUNIT cards.

---

*E. H. Means, C. L. Phillips, and S. E. Young, "BALFRAM User Manual for the Staff of the Commander in Chief Pacific," Stanford Research Institute, Menlo Park, CA 94025, Technical Note NWRC-TN-52, September 1974.

2. PROGRAM CONTROL CARD (6I5,20I2)

A Program Control Card must immediately follow the Title Card for each BALFRAM scenario. The Program Control Card provides options for user control of the scenario processing:

(1) FIELD 1 (columns 1-5) (I5) states how many nodes (i.e., the highest node number) are in the NODH file (file 20) for scenario geography. This number must not exceed 80.

(2) FIELD 2 (columns 6-10) (I5) contains the code, listed below, for the default fight law, which is implemented when a fight law is not specified on the applicable BTLENODE card. (See Appendix B of the User Manual for discussion of fight laws.)

Code	Law
Blank or 1	Differential square fight law.
2	Differential linear fight law.
3	Differential mixed fight law: BLUE suffers attrition by the differential square fight law, RED suffers attrition by the differential linear fight law.
4	Differential mixed fight law: RED suffers attrition by the differential square fight law, BLUE suffers attrition by the differential linear fight law.

(3) FIELD 3 (columns 11-15) (I5) sets the maximum number of battle steps to be performed. This limits the effects of possible loops in the calculations. It also enables the analyst to limit the scenario play for test purposes.

(4) FIELD 4 (columns 16-20) (I5) contains a code, listed below, that governs the printout of scenario results.

Code	Printout
0	Complete battle history, FEBA movement history, logistic interdiction history, and summaries.
1	Summaries only.

Code	Printout
2	FEBA movement history, logistic interdiction history, battle node information, and summaries.
3	Summaries and interpreted descriptors for each sensitivity analysis iteration.
4	Summaries plus normalized attrition factors for linear nonhomogeneous battle.

(5) FIELD 5 (columns 21-25) (I5) sets the number of battle steps to be performed before the battle history printout begins. If this field contains a zero or blank, the printout will begin immediately. If this field is negative (i.e., -n), the battle history is printed only every nth step.

(6) FIELD 6 (columns 26-30) (I5) contains blank or zero if the geography of the scenario--distance between nodes, minimum distance between any two nodes, and the matrix of next nodes in the path of minimum distance--is to be read in from file 20. Any other values are ignored.

(7) FIELDS 7 to 16 (columns 31-50) (10I2) consist of a user supplied program logic indicator array that is presently unused.

(8) FIELDS 17 to 26 (columns 51-70) (10I2) are indicators for selecting processing options. A blank or 0 in an option field has no effect; a 1 has the following effect:

Field	Columns	Processing Option
17	51-52	Write variables on file 1 after each event step for subsequent plotting or other processing.
18	53-54	Print battle history for an event only if it is not the result of a game time increment (i.e., print only nonroutine events).
19 to 26	55-70	Not presently used.

3. PARAMETER CHANGE CARDS

The single PARMCHNG Card in the original BALFRAM has been augmented by three new cards for changing parameter values at specified times during a BALFRAM scenario. The individual cards differ in the combinations of parameter codes, new values, and units on each card type. A maximum of 100 parameter changes is permitted in a scenario. Each of the multiple changes on a single PVCHANGE, UVCHANGE, and UNCHANGE card is counted as one parameter change. In contrast to the original BALFRAM, all parameter change reset values are subject to adjustment by FRCRATIO and RANDMSEQ descriptors, if appropriate. When changing NODEPROP parameters, see the instructions associated with the appropriate field for special conditions:

(1) PARMCHNG Card (2A4, 2X, A4, 1X, 2I5, 2F10.5). The PARMCHNG (parameter change) card resets a single parameter for a single unit to a new value at a given time during the BALFRAM scenario:

(a) FIELD 1 (columns 1-8) (2A4) contains 'PARMCHNG'.

(b) FIELD 2 (columns 11-14) (A4) contains 'BLUE' or 'RED' to denote the side to which the parameter change applies.

(c) FIELD 3 (columns 16-20) (I5) contains a code, listed below, that indicates which parameter is to be changed:

Code	Parameter
1	Increment to unit order of battle (Field 5, UNITSPEC card). Note that the increment must be multiplied by the sortie rate before entry (if applicable).
2	Not used.
3	ICE (Field 8, UNITSPEC card).
4	Mobility factor (Field 10, UNITSPEC card).
5	Nodal objective of unit (Field 12, UNITSPEC card).
6	Not used.
7	Exogenous firepower target node (Field 13, UNITSPEC card).
8	Disengagement code (Fields 5,7,9,..., GUERILLA card).
9	Defeat criterion--actual order of battle, not percent (see Field 7, UNITSPEC card).
10	Nodal location of units used to compute exogenous firepower (Field 4, NODEPROP card).
11	Target node for exogenous firepower (Field 5, NODEPROP card).

Code	Parameter
12	Node or unit for additional code depending on value in Field 7 of NODEPROP card (Field 6, NODEPROP card).
13	Exogenous firepower computation code (Field 7, NODEPROP card).

(d) FIELD 4 (columns 21-25) (I5). If Field 3 is 9 or less, Field 4 contains the identifier of the unit that is to have a parameter changed. If Field 3 is 10 or more, Field 4 contains the sequential position of the NODEPROP card to be changed with respect to all NODEPROP cards in the input.

(e) FIELD 5 (columns 26-35) (F10.5) contains the time at which the parameter is to change.

(f) FIELD 6 (columns 36-45) (F10.5) contains the new value of the parameter.

(2) PVCHANGE card (2A4, 2X, A4, 1X, F10.5, I5, 4 (I3, F7.2)). The PVCHANGE (parameter and value change) card resets multiple parameters for a single unit or NODEPROP card at a given time in the BALFRAM scenario:

(a) FIELD 1 (columns 1-8) (2A4) contains 'PVCHANGE'.

(b) FIELD 2 (columns 11-14) (A4) contains 'BLUE' or 'RED' to denote the side to which the parameter change applies.

(c) FIELD 3 (columns 16-25) (F10.5) contains the time at which the parameters are to change.

(d) FIELD 4 (columns 26-30) (I5). If Field 5 is 9 or less, Field 4 contains the identifier of the unit that is to have a parameter changed. If Field 5 is 10 or more, Field 4 contains the sequential position of the NODEPROP card to be changed with respect to all NODEPROP cards in the input.

(e) FIELDS 5,7,9, and 11 (columns 31-33, 41-43, 51-53, 61-63) (I3) contain a code that indicates which parameter is to be changed. The codes are same as for Field 3 of the PARMCHNG card, described in 3(1), above. All codes on a single PVCHANGE card must be either 9 or less or 10 or greater to preserve the correspondence with Field 4.

(f) FIELDS 6,8,10, and 12 (columns 34-40, 44-50, 54-60, 64-70) (F7.2) contain the new values of the parameter. See UNITSPEC and NODEPROP descriptor discussions in the User Manual for applicable special conditions for these values.

(3) UVCHANGE card (2A4, 2X, A4, 1X, F10.5, I5, 4 (I3, F7.2)). The UVCHANGE (unit and value change) card resets a single parameter on multiple UNITSPEC or NODEPROP cards with new values at a given time in the BALFRAM scenario:

(a) FIELD 1 (columns 1-8) (2A4) contains 'UVCHANGE.'

(b) FIELD 2 (columns 11-14) (A4) contains 'BLUE' or 'RED' to denote the side to which the parameter change applies.

(c) FIELD 3 (columns 16-25) (F10.5) contains the time at which the parameters are to change.

(d) FIELD 4 (columns 26-30) (I5) contains a code that indicates which parameter is to be changed. The codes are the same as for the PARMCHNG card, listed in 3(1), above.

(e) FIELDS 5,7,9, and 11 (columns 31-33, 41-43, 51-53, 61-63) (I3). If Field 4 is 9 or less, Field 5,7,9, and 11 contain the identifier of a unit that is to have a parameter changed. If Field 4 is 10 or more, Field 5,7,9, and 11 contain the sequential position of the NODEPROP card to be changed with respect to all NODEPROP cards in the input.

(f) FIELDS 6,8,10, and 12 (columns 34-40, 44-50, 54-60, 64-70) (F7.2) contain the new values of the parameter. See UNITSPEC and NODEPROP descriptor discussions in the User Manual for applicable special conditions for these values.

(4) UNCHANGE card (2A4, 2X, A4, 1X, F10.5, I5, 3X, F7.2, 10I3). The UNCHANGE (unit change) card resets a single parameter on multiple UNITSPEC or NODEPROP cards to a single value at a given time in the BALFRAM scenario:

(a) FIELD 1 (columns 1-8) (2A4) contains 'UNCHANGE.'

(b) FIELD 2 (columns 11-14) (A4) contains 'BLUE' or 'RED' to denote the side to which the parameter change applies.

(c) FIELD 3 (columns 16-25) (F10.5) contains the time at which the parameters are to change.

(d) FIELD 4 (columns 26-30) (I5) contains a code that indicates which parameter is to be changed. The codes are the same as for Field 3 of the PARMCHNG card, described in 3(1), above.

(e) FIELD 5 (columns 34-40) (F7.2) contains the new values of the parameter to be used for all units or NODEPROP cards. See the UNITSPEC and NODEPROP descriptor discussions in the User Manual for the applicable special conditions for this value.

(f) FIELDS 6ff. (columns 41-43, 44-46, ..., 68-70). If Field 4 is 9 or less, Fields 6ff. contain the identifier of the unit that is to have the parameter changed. If Field 4 is 10 or more, these fields contain the sequential position of the NODEPROP card to be changed with respect to all NODEPROP cards in the input.

4. STOPBTLE CARD (2A4, 2X, A4, 1X, I5, 20I3/10I3)

STOPBTLE (stop battle) cards cause the scenario play to terminate when: all the units designated on a STOPBTLE card fall below their respective defeat criteria (Field 7, UNITSPEC card); any one of the units designated on the STOPBTLE card attains its geographic objective (Field 12, UNITSPEC card); or simulated game time exceeds the designated time. A maximum of 5 STOPBTLE cards is permitted:

(1) FIELD 1 (columns 1-8) (2A4) contains 'STOPBTLE.'

(2) FIELD 2 (columns 11-14) (A4) contains 'BLUE' or 'RED' to denote the side on which the designated units fight. Use either BLUE or RED for time limit.

(3) FIELD 3 (columns 16-20) (I5) indicates how many values are entered in Fields 4ff. (must be 30 or less).

(a) For defeat criteria, enter the number of units governed by this card.

(b) For attaining objective, enter the number of units governed by this card plus 1.

(c) For time limit, enter 2.

(4) FIELD 4 (columns 21-23) (I3) contains a unit identifier or a code indicating alternate use of the STOPBTLE card.

(a) For defeat criteria, enter the unit identifier of the first unit governed by this card.

(b) For attaining objective, enter -2 (negative 2).

(c) For time limit, enter -1 (negative 1).

(5) FIELD 5 (columns 24-26) (I3) contains a unit identifier or the time limit for stopping the battle.

(a) For defeat criteria, enter the unit identifier of the second unit governed by this card (if any).

(b) For attaining objective, enter the unit identifier of the first unit governed by this card.

(c) For time limit, enter the time limit for ending the game rounded to the nearest integer and entered without a decimal point.

(6) FIELDS 6ff. (columns 27-29, 30-32, ..., 78-80, 1-3, ...) (I3) contains the unit identifiers for the remaining units in the list for defeat criteria or attaining objective. A continuation card may be used if required, format (I3), beginning in column 1. If the entry in Field 3 is 20, a blank continuation card is required even though all the units are listed on the basic card.

5. PROASIGN CARD SET

The PROASIGN (proportional assignment) card sets apportion the forces in a theater of war. At each step in the model calculations, the total surviving order of battle of a set of units is redistributed among the units according to specified rules. Two sets of rules are available with the PROASIGN card. One set is described in the BALFRAM User Manual. This section documents an additional set added subsequent to publication of the User Manual. A maximum of twenty total PROASIGN cards is permitted.

With this alternate use of the PROASIGN Card, two sets of units are used--Set A and Set B. Set A is used in a fashion similar to the set of units in the other use of the PROASIGN card. The BALFRAM program computes two control values--the sum of the remaining order of battle of the units in Set A and the product of a constant (last value entered in Fields 12 to 18) multiplied by the remaining order of battle of the unit in Set B. The program then apportions the lesser of these two values to the units in Set A according to the factors specified (in Fields 12 to 18):

(1) Card 1 for PROASIGN Set (2A4, 2X, A4, 2I2, 7I3, 1X, 7F5.1):

(a) FIELD 1 (columns 1-8) (2A4) contains 'PROASIGN.'

(b) FIELD 2 (columns 11-14) (A4) contains 'BLUE' or 'RED' to denote the side whose forces are to be redistributed.

(c) FIELD 3 (columns 15-16) (I2) contains a 2 to have the program read Card 2 of this PROASIGN card set.

(d) FIELD 4 (columns 17-18) (I2) indicates how many units are in the two sets. This number must be greater than 1, but must not exceed 7. Set B must contain exactly one unit.

(e) FIELDS 5 to 11 (columns 19-21, 22-24, ..., 37-39) (I3) contain the identifiers of the units among which the order of battle is to be apportioned (Set A) and the control unit (Set B), given in successive fields of three columns each. Units to which explicit quantities of the order of battle are to be assigned should be input according to priority and should precede units that are to receive proportional assignments. If the specified order of battle is insufficient for all of the explicit quantity assignments, full assignments will be made in the order in which these units are input until the surviving order of battle is exhausted. Units must be entered for Set A first followed by the Set B unit.

(f) FIELDS 12 to 18 (columns 41-45, 46-50, ..., 71-75) (F5.1) contain the apportionment factors applied to the units in Fields 5 to 11 (Set A) and the constant multiplier for the unit in Set B, respectively. If the numbers for Set A are negative, the absolute value is the explicit quantity of the order of battle assigned to the corresponding unit in Fields 5 to 11. If the numbers in these fields are

positive, they indicate the fraction of the total force remaining, after the previous assignments, to be allocated to the corresponding unit. (For example, entries of 0.75, 0.5, and 1.0 mean that 75% of the total order of battle is to be assigned to the first unit, 50% of the remaining order of battle to the second unit, and all the remaining order of battle to the third unit.)

(2) Card 2 for PROASIGN Set (40X, F5.1)

(a) FIELD 1 (columns 41-45) (F5.1) contains the negative number of units in Set A. The negative sign is the indicator for this alternate use of the PROASIGN Card Set.

6. UNITSPEC, LGINTDIC, and SUMUNIT Card Changes

6.1 UNITSPEC Card Change

The UNITSPEC card describes the forces in the BALFRAM scenario. A single field on the UNITSPEC card has been changed to require input of an initial sortie rate for the order of battle resources. The following paragraph should be inserted in the User Manual to describe this change.

(6) FIELD 6 (columns 31-35) (F5.1) indicates the initial sortie rate for the forces provided in Field 5 above. The sortie rate is the number of engagements performed per scenario time unit for each force element. The initial order of battle used in the scenario is the product of Field 5 and Field 6. If a sortie rate does not apply to the forces in Field 5, 1.0 is entered in Field 6.

6.2 LGINTDIC Card Change

Field 3 on the LGINTDIC (logistic interdiction) card specifies the number of nodes for resupply. The maximum number of nodes has been increased from 10 to 20.

6.3 SUMUNIT Card Change

Field 5 on the SUMUNIT (summed units) card specifies the number of units that are to have their parameters aggregated. The maximum number of units has been increased from 10 to 20.

Appendix B

BLOCK DIAGRAMS OF BALFRAM CONTROL ROUTINES

43

11

Appendix B

BLOCK DIAGRAMS OF BALFRAM CONTROL ROUTINES

To document more fully the logical working of the BALFRAM program, this appendix presents block diagrams of the major control routines. These block diagrams show the flow of program processing at a macro-level of detail. Individual FORTRAN statements that perform particular functions are grouped together and described by the function performed. The logical branches that control the execution of the major program sections are also shown. At this level of detail the important program construction and logical features are evident without the confusing burden of detailed variable processing.

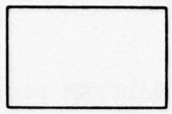
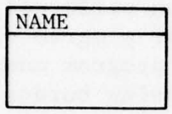
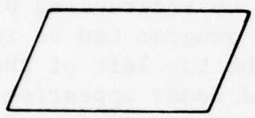
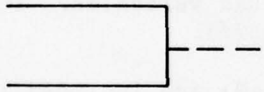
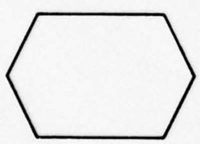
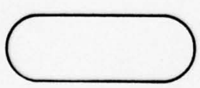
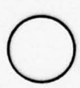
Table B-1 defines all the symbol shapes used in the block diagrams. The block diagrams are keyed to the FORTRAN program by referencing program statement numbers so that specific code in the program can be readily identified. The statement numbers are located on the top left of the symbol describing the associated logic. Capitalized names appearing within symbols are program variable names and are used to aid in understanding the detailed processing of the subprograms. In many cases, the same processing is performed on the Blue then the Red variables. In such cases, the variable names are separated by a slash (/).

Block diagrams (presented as Figures B-1 to B-9) are included in this appendix for the following subprograms:

- INILT
- SURFGN
- RANDM
- TSMO
- BATTLE
- INVBAT
- LINBAT
- INVLB
- ADCP5

Table B-1

BLOCK DIAGRAM SYMBOL DEFINITIONS

Symbol*	Definition
	Process symbol--executing a defined operation or group of operations resulting in a change of value, form, or location.
	Process performed by a subroutine or function named NAME.
	Input or output function.
	Annotation function--addition of descriptive comments or explanatory notes.
	Decision function--switching operation that determines which alternative path is to be followed.
	Terminal symbol--start or end of a subprogram.
	Connector symbol--a junction in the line of flow. Two connectors are used to represent continued flow when flow would be broken by flow chart limitation.

*Numbers on upper left side of symbol (if present) indicate the FORTRAN statement number near or at the point in the program where the described action is performed.

Capitalized names appearing within symbols refer to variables within the program.

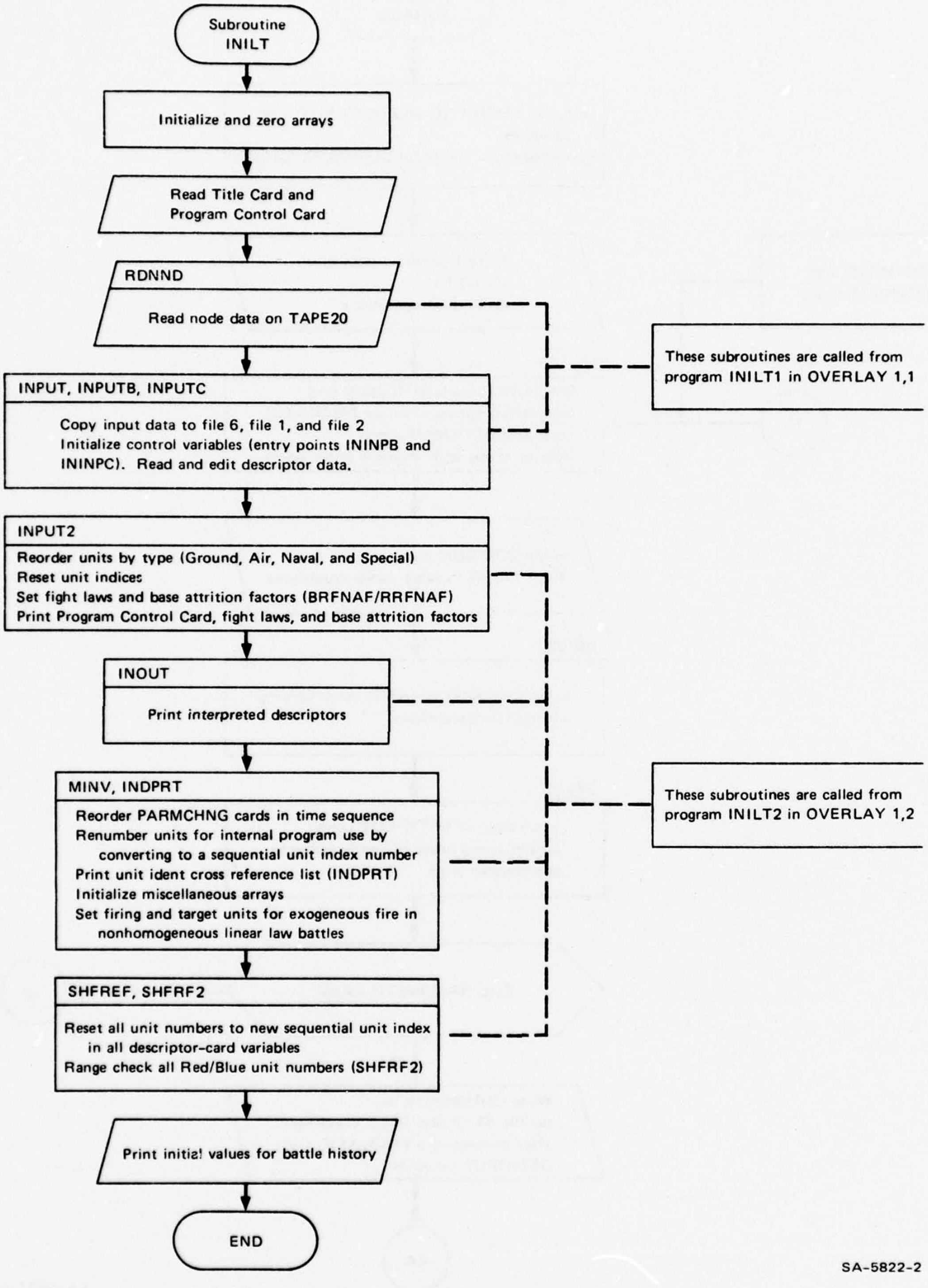


FIGURE B-1 SUBROUTINE INILT BLOCK DIAGRAM

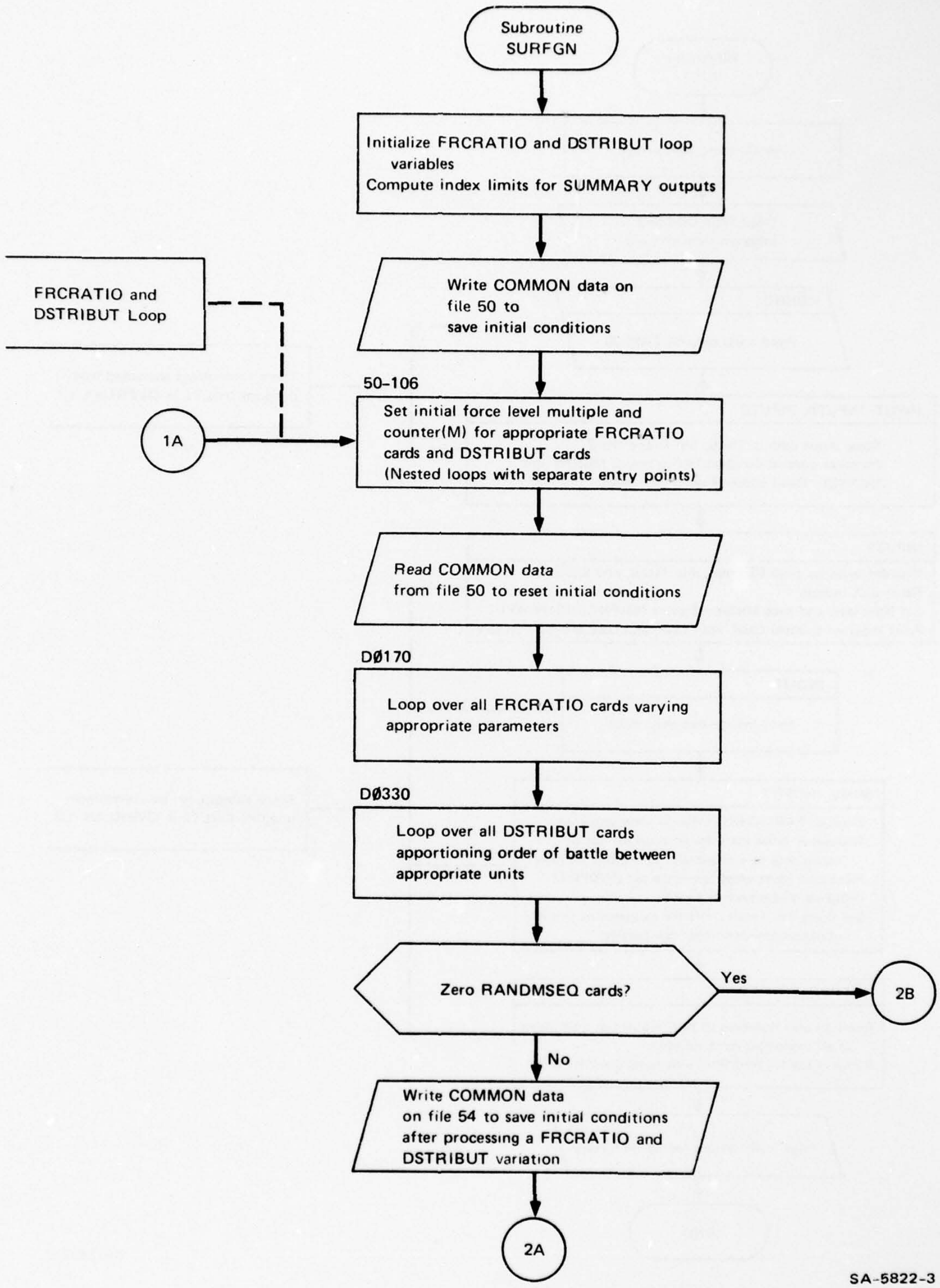


FIGURE B-2 SUBROUTINE SURFGN BLOCK DIAGRAM

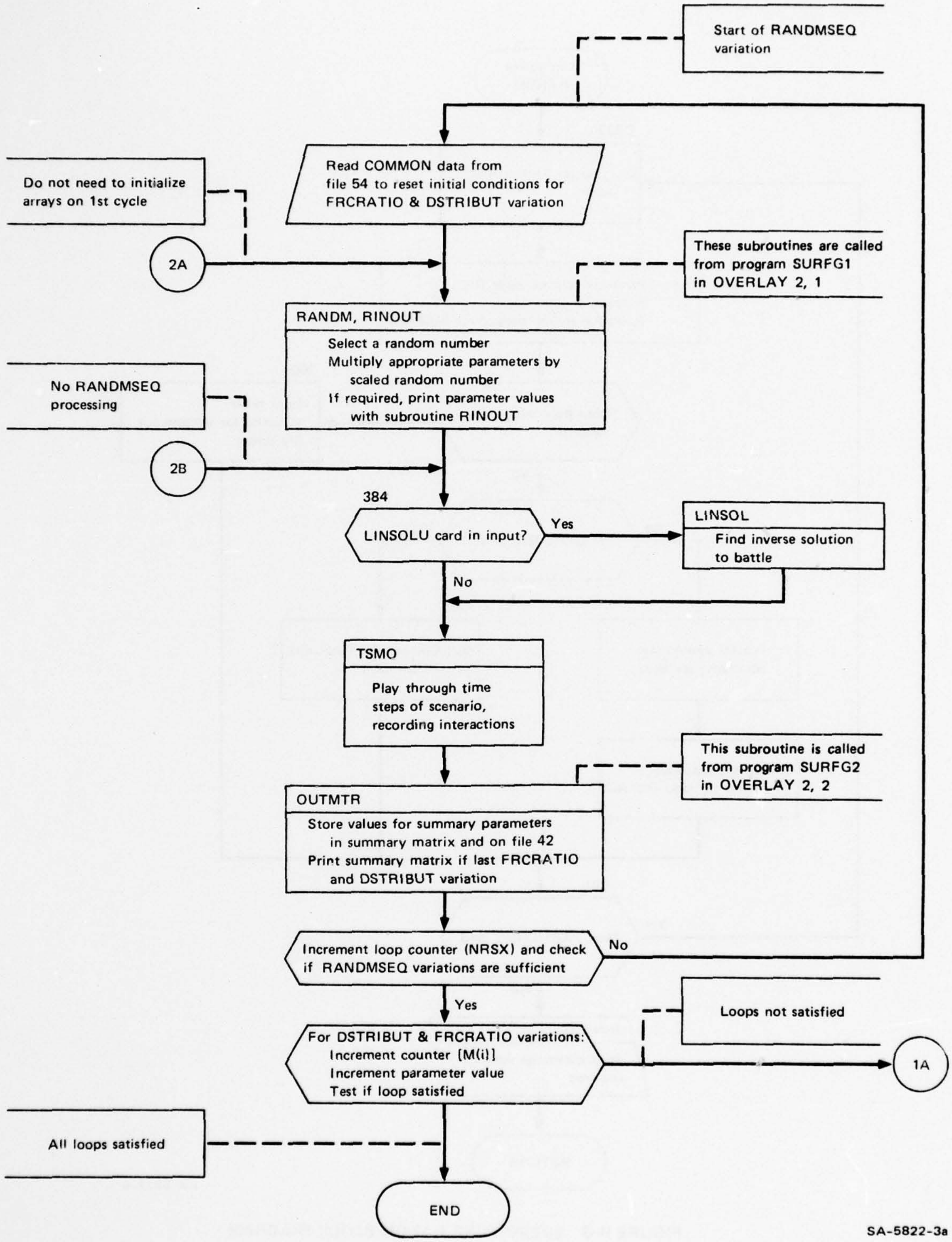


FIGURE B-2 SUBROUTINE SURFGN BLOCK DIAGRAM (Concluded)

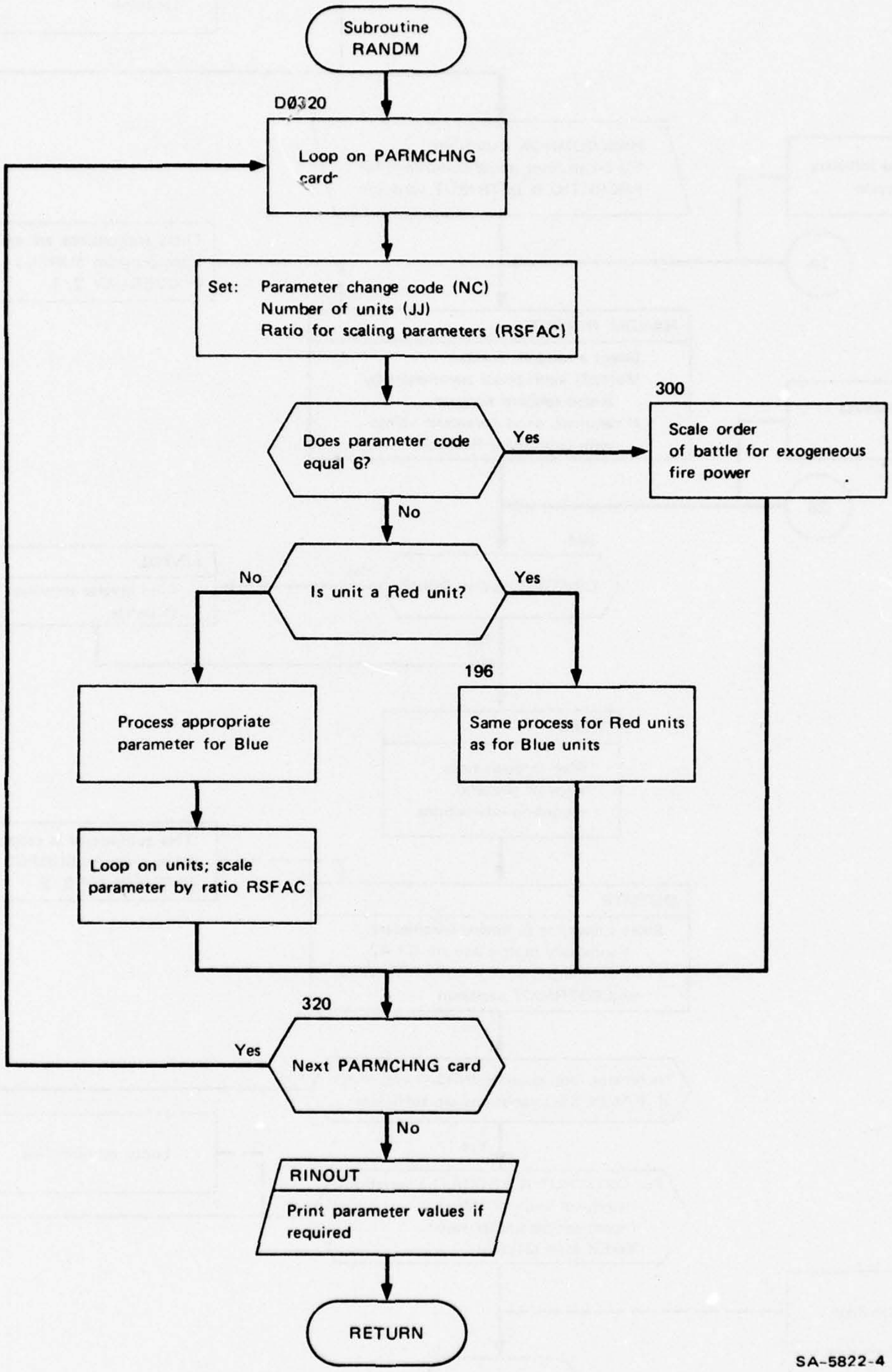


FIGURE B-3 SUBROUTINE RANDM BLOCK DIAGRAM

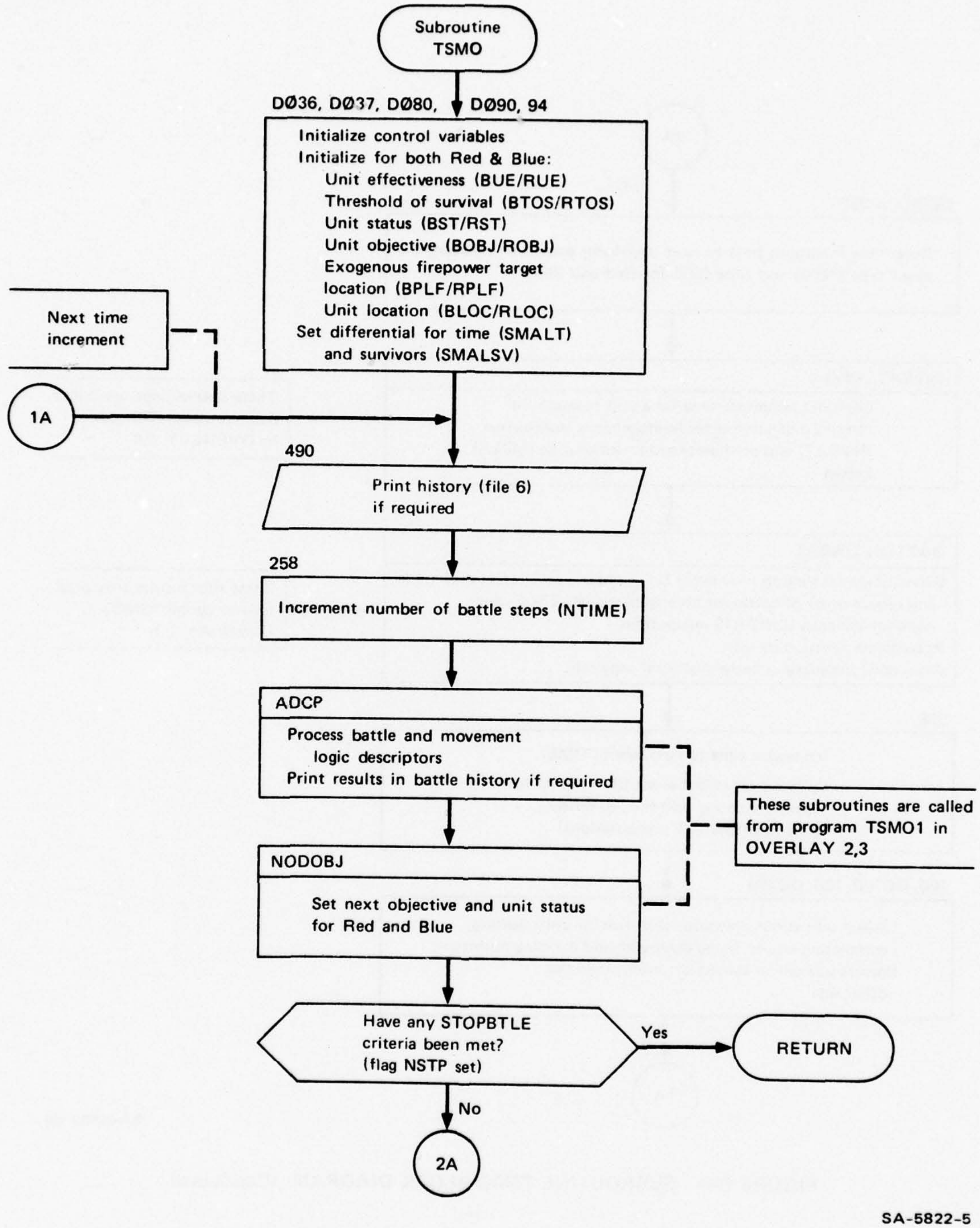
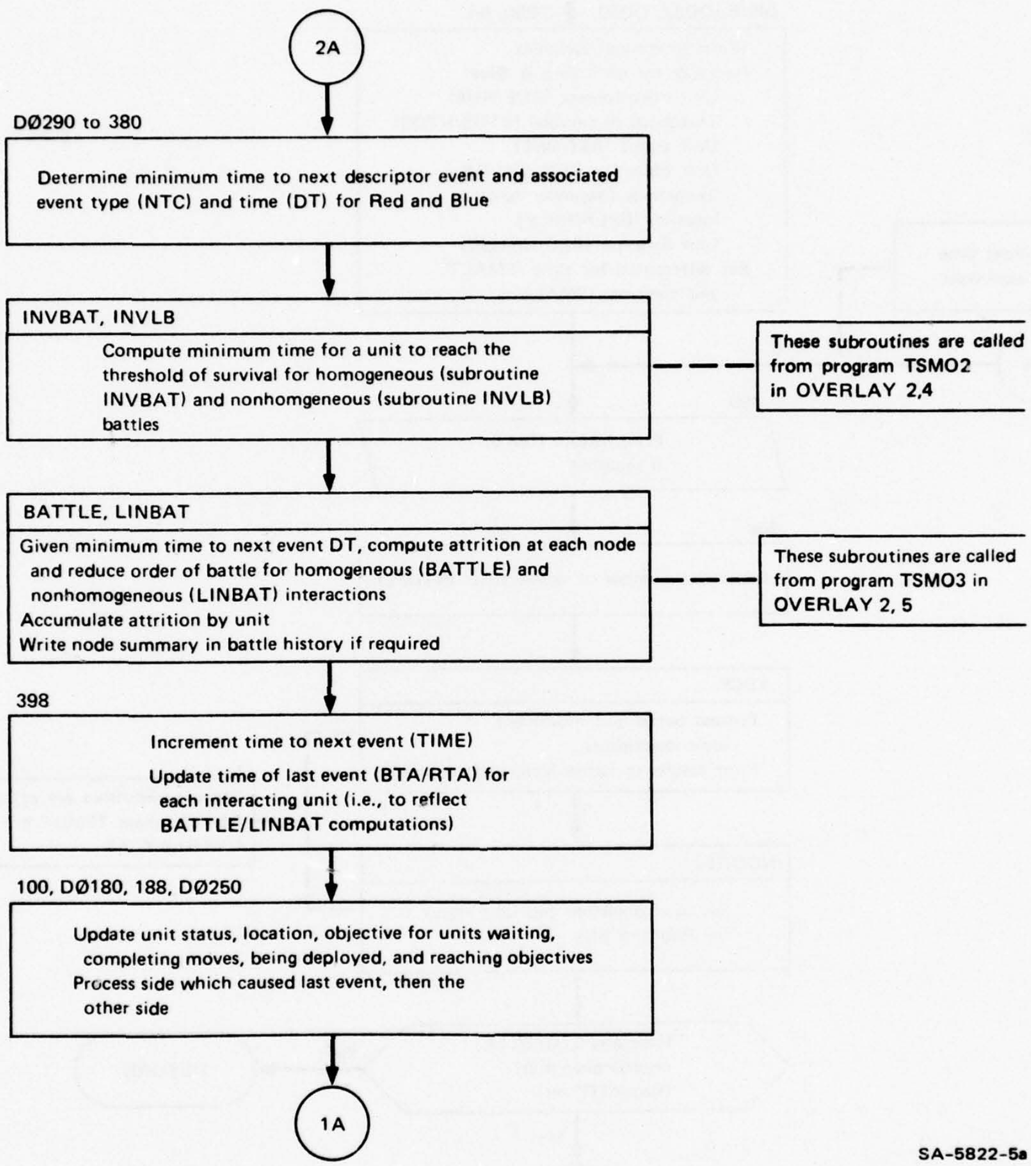


FIGURE B-4 SUBROUTINE TSMO BLOCK DIAGRAM



SA-5822-5a

FIGURE B-4 SUBROUTINE TSMO BLOCK DIAGRAM (Concluded)

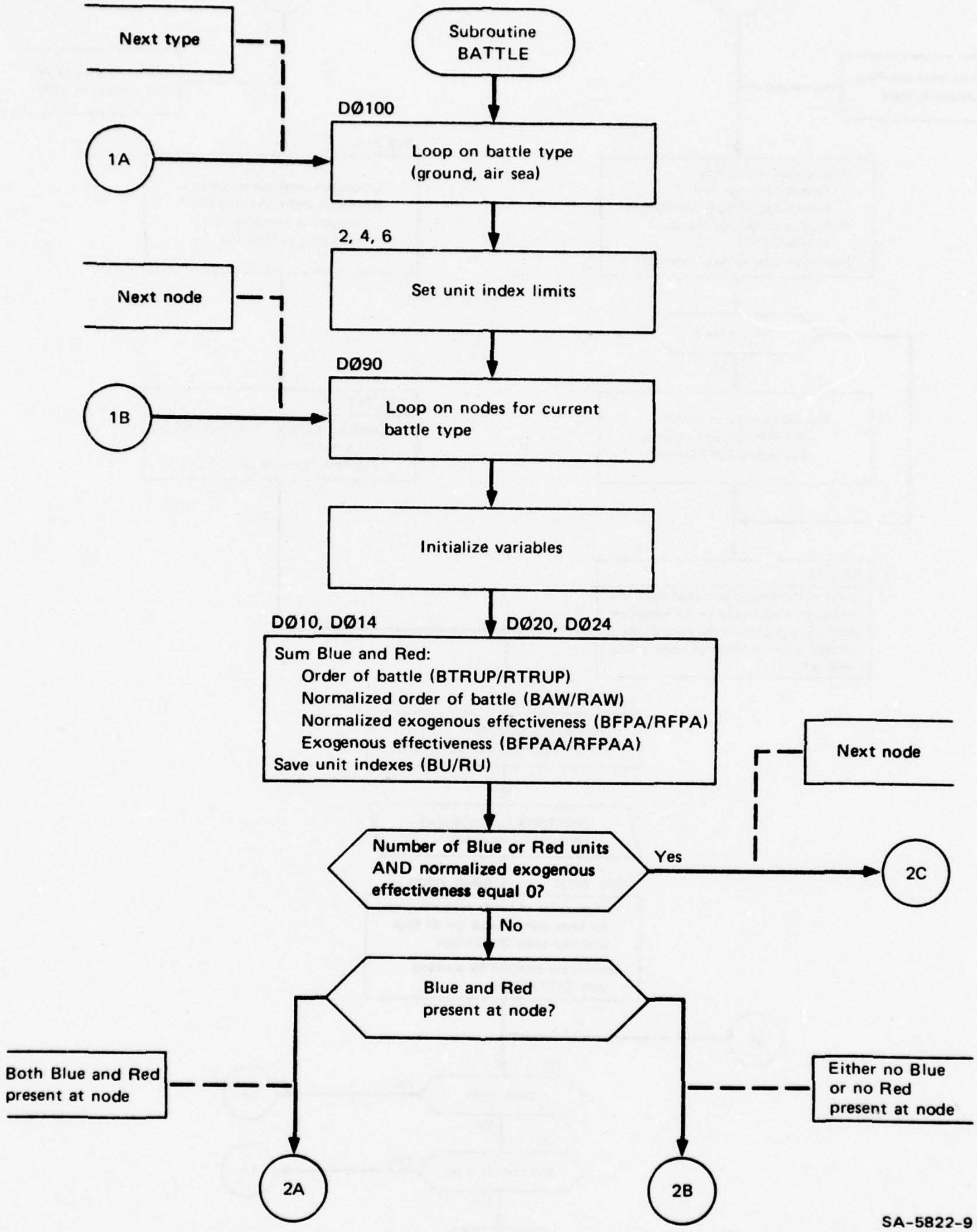


FIGURE B-5 SUBROUTINE BATTLE BLOCK DIAGRAM

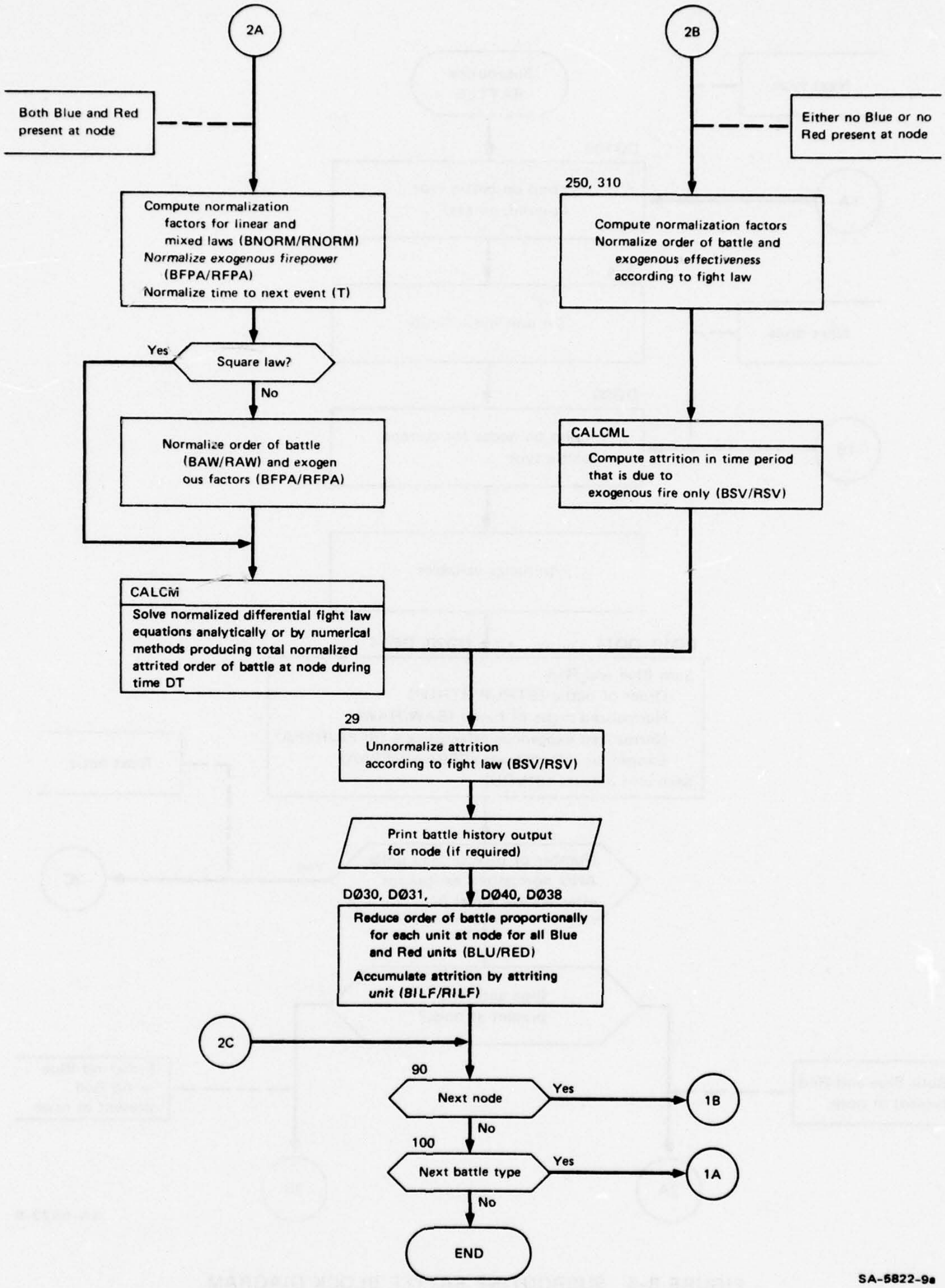


FIGURE B-5 SUBROUTINE BATTLE BLOCK DIAGRAM (Concluded)

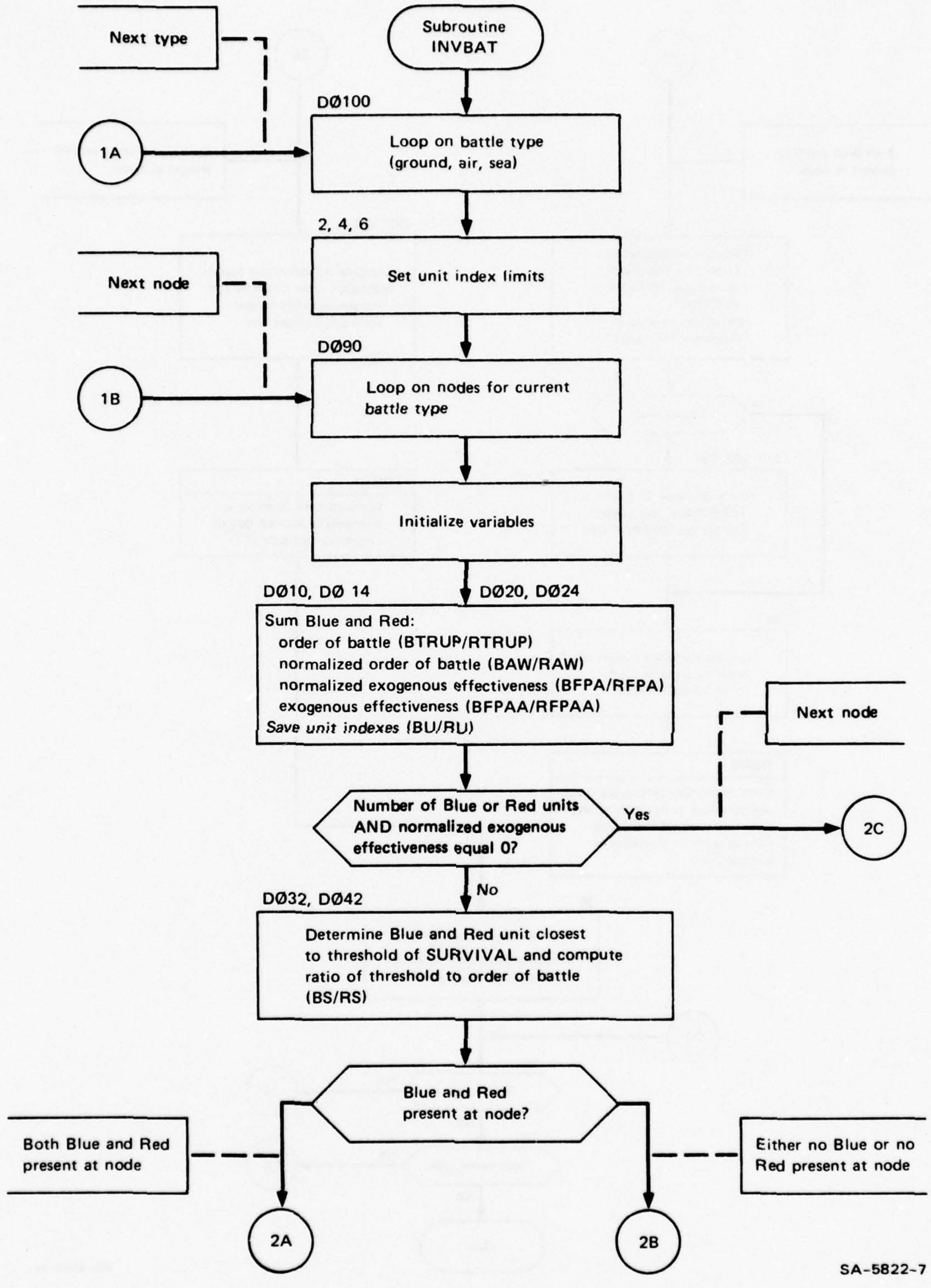


FIGURE B-6 SUBROUTINE INVBAT BLOCK DIAGRAM

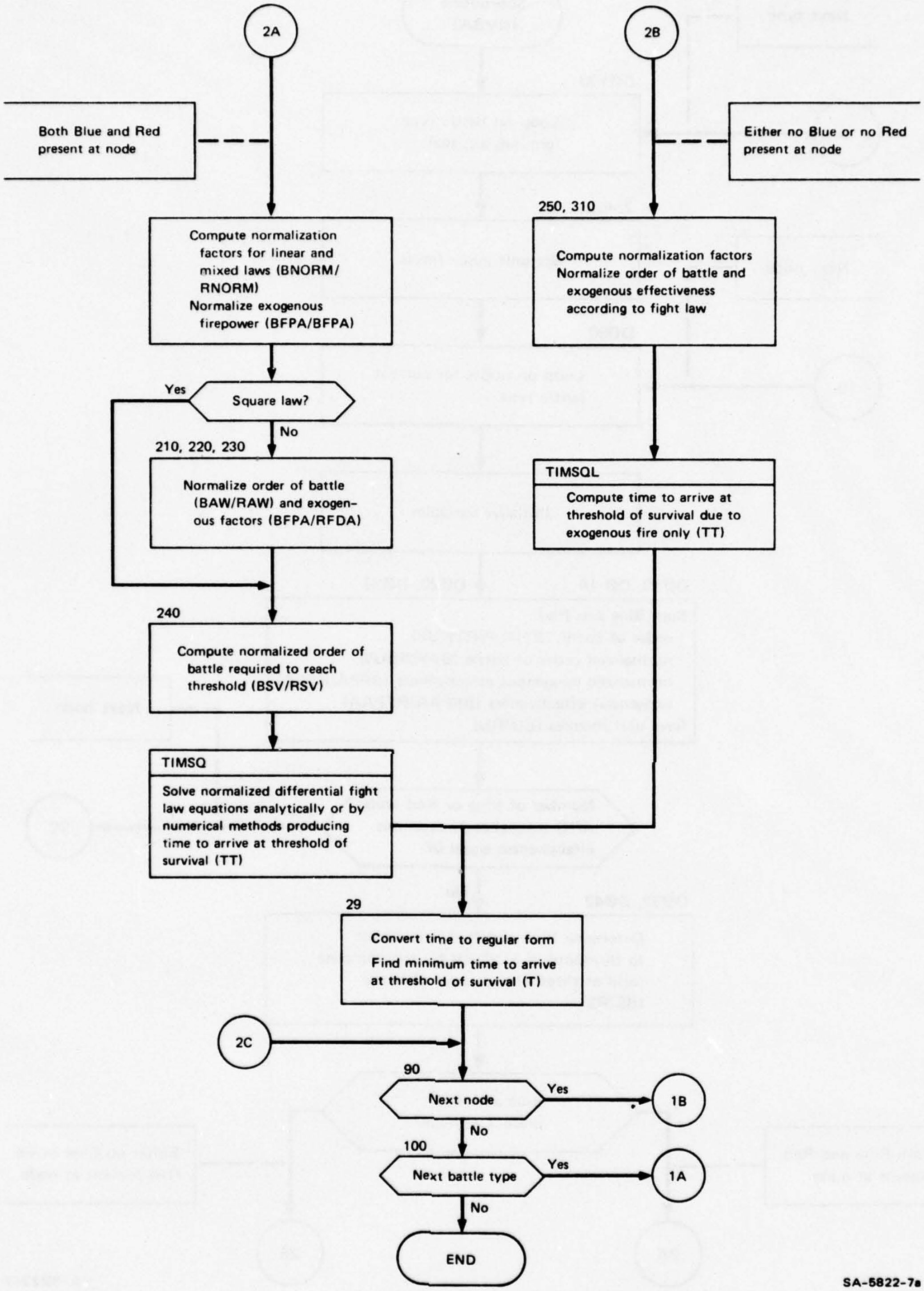


FIGURE B-6 SUBROUTINE INVBAT BLOCK DIAGRAM (Concluded)

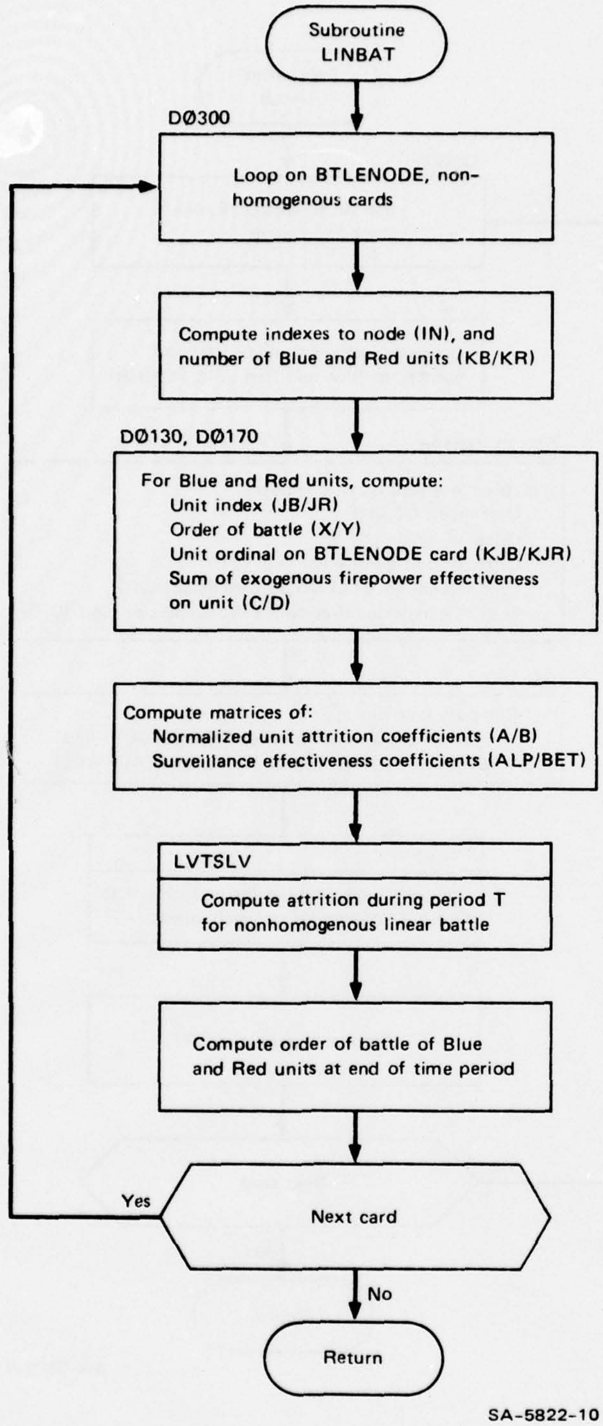


FIGURE B-7 SUBROUTINE LINBAT BLOCK DIAGRAM

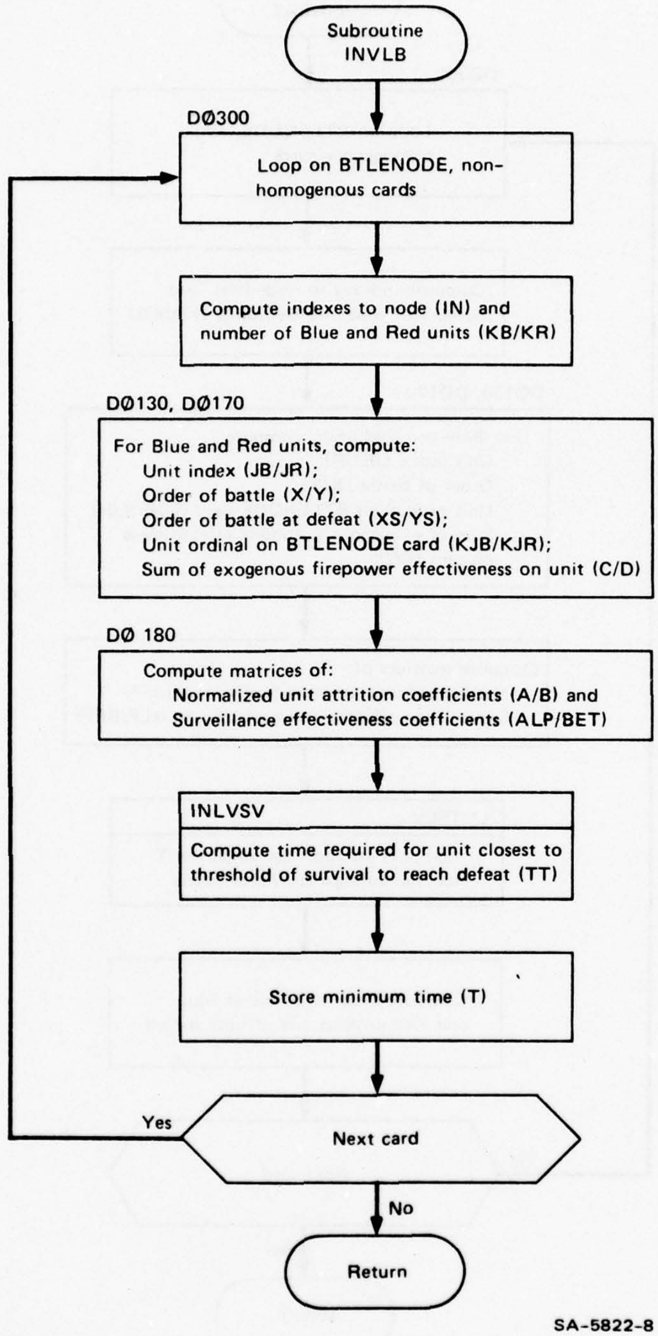


FIGURE B-8 SUBROUTINE INVLB BLOCK DIAGRAM

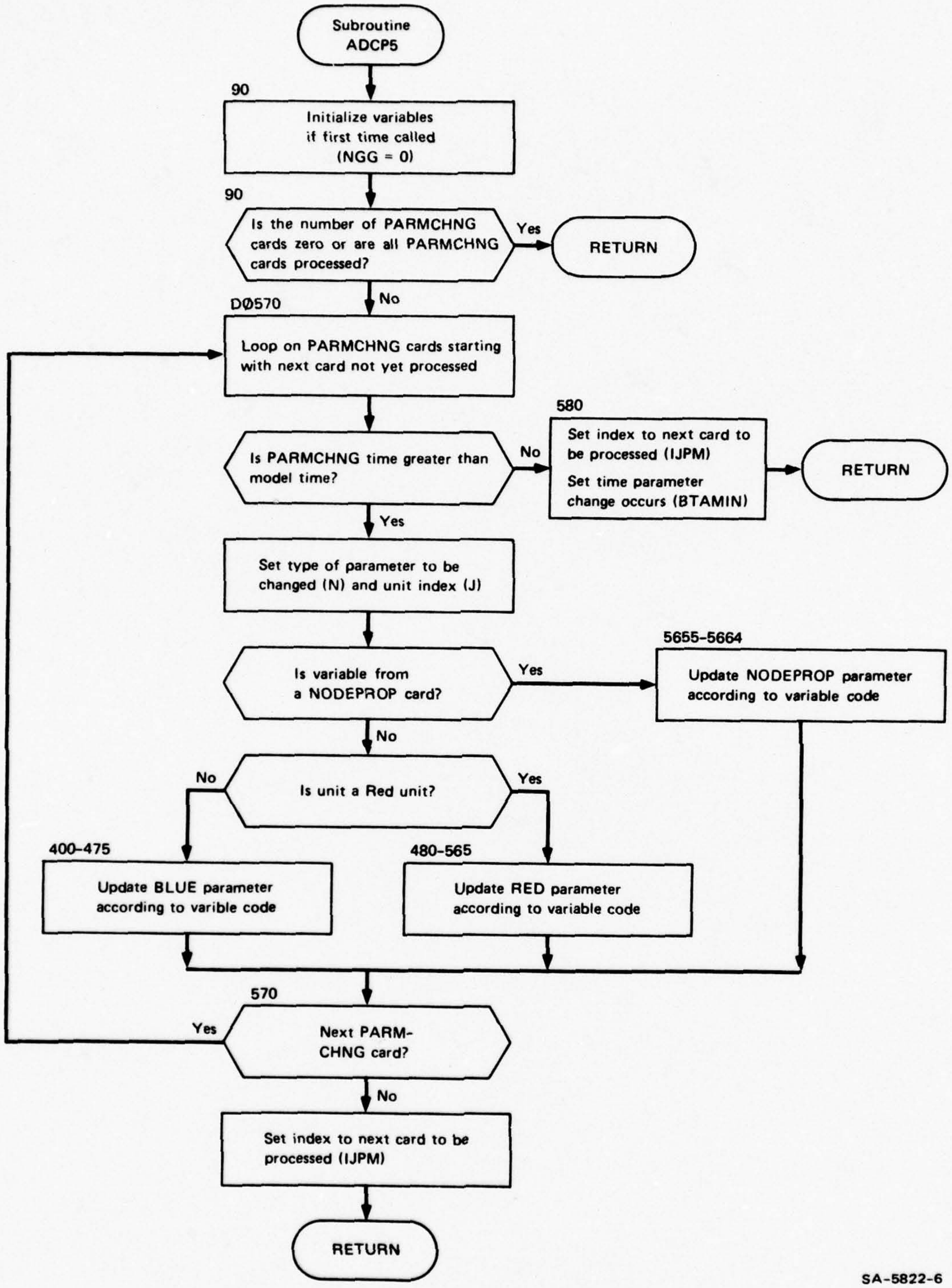


FIGURE B-9 SUBROUTINE ADCP5 BLOCK DIAGRAM

Precediing Page Blank - Not Filmed

Appendix C

VARIABLE DEFINITIONS

59

Preceding Page Blank - NOT FILMED

Appendix C

VARIABLE DEFINITIONS

The information content of variables is critical to understanding and changing the features of a program. A dictionary of the variables in each COMMON block of BALFRAM has been assembled and updated during this research and is included in this appendix.

The BALFRAM program uses numerous dimensioned variables or arrays to store input data and intermediate results. Variables are assembled into COMMON blocks according to their usage. Generally, variables associated with a particular descriptor are organized into the same COMMON block. To preserve the correspondence of variables and their COMMON blocks, the variable definitions are also organized by COMMON block. All variables in each COMMON block are defined, their size (DIMENSION) given, and the source of the data is provided for cross reference. Table C-1 is an alphabetical guide to the page location of each COMMON block definition.

Table C-2 provides the current dimensions allowed for each descriptor type, the symbol for each dimension that is used in the tables in this appendix, and the COMMON where associated data is stored. The remaining Tables C-3 to C-27 provide the definitions of the variable parameters for each COMMON. The data origin entry specifies the source of the data in that variable. If the data origin is a specific field on one of the descriptor cards, refer to the BALFRAM User Manual for further information. If the entry is a subroutine name, refer to that subroutine. An entry enclosed in parentheses means that at least one step of processing is involved between the source shown and the storage array.

This appendix should replace Appendix D in the BALFRAM Programmer Maintenance Manual.

Table C-1

GUIDE TO COMMON BLOCK TABLES

Common Name	Page Number
BLUE	68
CALCMC	69
DCMN	69
E	70
F	73
FB	73
FBA	74
G	74
H	75
L	76
LG	77
LSOLA	77
NLOG	78
OL11	78
OVL22	79
OVL23	79
OVL245	79
PATCH	78
PLAN56	79
PLN	80
R	81
RED	81
SUMRY	82
SUNIT	82
TEMP	83

Table C-2

DESCRIPTOR DIMENSION LIMITS

Descriptor Type	Maximum Dimension Allowed		COMMON Block(s) in Which Used
	Number	Name	
UNITSPEC	120	$NU	BLUE,RED,DCMN
NODEPROP	80	$NP	E
FRCRATIO	6	$FR	F
ADVANCE	40	$AD	E
RETREAT	40,20	$RT,$RT2	E
DSTRIBUT	12,20	$DR,$DR2	H
SUMMARY	40,50	$SM,$SM2	SUMRY
BTLENODE			
a. Homogeneous	50	$BH	CALCM
b. Nonhomogeneous	10,20,20	$BN,$BN2,$BN3	L
OBJCTADV	40,20	$OA,$OA2	E
NODATFAC	20	$NA	E
PROASIGN	20	$PR	G
STRTEGRT	40	$SR	G
TIMEADV	40	$TA	E
RANDMSEQ	20,30	$RS,$RS2	R
SUMUNIT	10,20	$SU,$SU2	SUNIT
REDEPLOY	10	$RE,$RE2,$RE3	E
CHASE	20	$CH	G
RENDEVOU	20	$RV	G
EXOGUNIT	40	$EX2	E
OPEXOGUN	10	$OP	E
LGINTDIC	5,20,20	$LG,$LG2,$LG3	LG
PARMCHNG	100	$PC	FB
STOPBTLE	5	$ST	E
LINSOLU	1	--	LSOLA
Number of Nodes	80	$ND	CALCM

Table C-3

COMMON BLOCK "BLUE" ARRAYS

Parameter	Meaning	Origin	Dimension
BLU (120)	Order of battle (number of units)	UNITSPEC-5	$NU
BFP (120)	Exogenous firepower effectiveness (components attritted per unit time)	UNITSPEC-6	$NU
BTOS (120)	Defeat criterion (in OB units)	UNITSPEC-7	$NU
BNAF (120)	Index of combat effectiveness (ICE)	UNITSPEC-8	$NU
BTFD (120)	Time of first deployment	UNITSPEC-9	$NU
BMF (120)	Mobility factor	UNITSPEC-10	$NU
* BLOC (120)	Node location	Subroutine TSMO	$NU
* BOBJT (120)	Node objective (ultimate)	UNITSPEC-12	$NU
* BOBJ (120)	Node objective (immediate)	Subroutine TSMO	$NU
* BST (120)	Status	Subroutines TSMO,NODOBJ	$NU
BTA (120)	Time of last discontinuous event	Subroutine TSMO	$NU
BRNF (120)	Square root of (ICE) normalized attrition factor ($\sqrt{\text{BNAF}}$)	Subroutine MINV	$NU
BTLF (120)	Not used		$NU
* BPLF (120)	Location of applied exogenous support	UNITSPEC-13	$NU
* BDLC (120)	Initial node location	UNITSPEC-11	&NU
* BRST (120)	Retreat status	GUERILLA-4ff.	$NU
BILF (120)	Cumulative attrition inflicted by unit	Subroutine BATTLE	$NU
A4 TYPEB (120)	Unit type (AIR, GRND, NAVL, SPEC)	UNITSPEC-3	$NU
BTH (120)	Survival ratio (fraction 0 to 1.0)	UNITSPEC-7	$NU
BUE (120)	Unit effectiveness — initialized to BNAF	Subroutine MINV	$NU
BFPF (120)	P Factor		$NU

* - Denotes integer variable having "REAL" name.

A4 - Denotes a 4 character per word holerith variable.

Table C-4

COMMON BLOCK "CALCMC" ARRAYS

Parameter	Meaning	Origin	Dimension
FN (3)	Number of battle nodes by homogeneous type	BTLENODE-3	3 {GRND = 1, AIR = 2, NAVL = 3}
FNA (3,50)	Homogeneous battle nodes by battle type	BTLENODE-4ff.	3 x $BH
IFLAW (80)	Differential fight laws	BTLENODE-Card 2	$NU
BFNAF (80)	Base attrition factors - BLUE (opposing force attritted per unit time)	BTLENODE Card 3	$NU
RFNAF (80)	Base attrition factors - RED	BTLENODE Card 4	$NU
BRFNAF (80)	Square root of BLUE base attrition factor	Subroutine INPUT2	$NU
RRFNAF (80)	Square root of RED base attrition factor	Subroutine INPUT2	$NU
ICALCM	Fight law for Node I [IFLAW(I)]	Subroutine INVBAT	1

Table C-5

COMMON BLOCK "DCMN" ARRAYS

Parameter	Meaning	Origin	Dimension
DISTPK (300) NODEPK (300)	Packed internode distances and indexes	Subroutine MDIST	
NEXTPK (1600)	Packed next node data	Subroutine MDIST	
NUNITB (120)	User BLUE unit identifier	UNITSPEC-4	$NU
NUNITR (120)	User RED unit identifier	UNITSPEC-4	$NU
INUNTB (120)	BLUE cross reference index (user ident to internal index)	Subroutine MINV	$NU
INUNTR (120)	RED cross reference index	Subroutine MINV	$NU
TIME	Simulation current time	Subroutine TSMO	
NOD	Number of nodes in data from file 20	Program Control Card	
NUMDST	Number of packed intermode distances	Subroutine RDNND	
NSKIP	Code controlling the printing of battle history	Program Control Card	
NB	Total number of BLUE units input	Subroutine INPUT	
NR	Total number of RED units input	Subroutine INPUT	
BTAMIN	Time to next event	Subroutine TSMO	
NSIZE	Set thresholds for survivors (BTOS) and TIME in TSMO		

Table C-5 (Concluded)

Parameter	Meaning	Origin	Dimension
NPRSKP	Code controlling frequency of battle history print	Program Control Card	
NUT	Length of COMMON BLUE and RED	Subroutine INILT	
NUNTB	Maximum number of RED units permitted	Subroutine INILT	
NUNTR	Maximum number of BLUE units permitted	Subroutine INILT	
NODET	Maximum number of nodes permitted	Subroutine INILT	
NBUG	Maximum number of battle steps permitted	Program Control Card	
NGG	Flag for initialization	Subroutine TSMO	
NPREP			
NBG1,NBG2,...	Index of start and end units in list of units by side and type	Subroutine INPUT2	

Table C-6

COMMON BLOCK "E" ARRAYS

Parameter	Meaning	Origin	Dimension
IDEPLY	Number of REDEPLOY descriptors		
IADN	Number of ADVANCE descriptors		
INPU	Number of NODEPROP descriptors		
IREN	Number of RETREAT descriptors		
ISTOPS	Number of STOPBTLE descriptors		
IOBN	Number of OBJCTADV descriptors		
IDEPUN (20,10)	Units which are to redeploy	REDEPLOY-Card 2 - 1st	20 x $RE
INOPFM (10)	Node from which units redeploy	REDEPLOY-4	$RE
INOPLY (10)	Node to which units redeploy	REDEPLOY-6	$RE
INOBJ (10)	Node which may suppress redeployment	REDEPLOY-7	$RE
TDEPLY (10)	Time delay at redeploy wait node	REDEPLOY-8	$RE
IEUN (30,10)	Units which might suppress redeployment	REDEPLOY-Card 2 - 2nd	30 x $RE
IADU (20,40)	Identifiers of units governed by ADVANCE card	ADVANCE-8ff.	20 x $AD
IADNOD (40)	Node to which advance moves	ADVANCE-5	$AD
IADFRM (40)	Node from which advance is made	ADVANCE-4	$AD

Table C-6 (Continued)

Parameter	Meaning	Origin	Dimension
IADF (40)	Advance criterion	ADVANCE-6	$AD
INPUN (80)	Unit identifier for exog. firepower (neg. if RED)	NODEPROP-3	$NP
INPNOD (80)	Node no. which triggers exogenous firepower	NODEPROP-4	$NP
INPOBJ (80)	Target node for exogenous firepower	NODEPROP-5	$NP
IREU (20,40)	Identifiers of units governed by RETREAT card	RETREAT-8ff.	20 x $RT
IRENOD (40)	Node to which retreat is made	RETREAT-5	$RT
IREFRM (40)	Node from which retreat is made	RETREAT-4	$RT
IREF (40)	Criterion node for retreat	RETREAT-6	$RT
ISTPU (30,5)	Identifiers of units goverened by STOPBTLE card	STOPBTLE-4ff.	$ST2 x $ST
NUMSTP (5)	Number of units governed by STOPBTLE card	STOPBTLE-3	$ST
IEDPUN (10)	Number of enemy units in set to suppress redeployment	REDEPLOY-9	$RE
INOWT (10)	Wait node for redeployment	REDEPLOY-5	$RE
IADJ (40)	Number of units on ADVANCE card (neg. if RED)	ADVANCE-3	$AD
IREJ (40)	Number of units on RETREAT card (neg. if RED)	RETREAT-3	$RT
IDENUM (10)	Number of units to be redeployed	REDEPLOY-3	$RE
CONTM (4,80)	Times at which exogenous type factors change	(NODEPROP-11,13)	4 x $NP
CONINP (4,80)	Exogenous firepower effectiveness scaled to OB fractions	(NODEPROP-10,12,14)	4 x $NP
INPTYP (80)	Code for computation of exogenous firepower	NODEPROP-7	$NP
IOBJ (40)	Number of units governed by OBJCTADV card (neg. if RED)	OBJCTADV-3	$OA
IOBU (20,40)	Identifiers of units to be advanced	OBJCTADV-6ff.	20 x $OA
IOBNOD (40)	Node to which units advance	OBJCTADV-5	$OA
IOBFRM (40)	Node from which units advance	OBJCTADV-4	$OA
INPOB2 (80)	Additional node for exogenous firepower	NODEPROP-6	$NP
NMNOD (20)	Node at which ICE adjustment is to be applied	NODATFAC-2	$NA
FLSC (20)	Force parameter ratio - trigger	NODATFAC-5	$NA
TFLSC (20)	Elapsed time criterion	NODATFAC-6	$NA
FLVL (5,2,20)	Force parameter ratios	NODATFAC-Card 2,3-1,4..	5 x 2 x $NA
BATF (5,2,20)	BLUE adjustment factors	NODATFAC-Card 2,3-2,5..	5 x 2 x $NA
RATF (5,2,20)	RED adjustment factors	NODATFAC-Card 2,3-3,6..	5 x 2 x $NA
FPDFAC (20)	Exogenous firepower modifications factor	NODATFAC-4	$NA

Table C-6 (Concluded)

Parameter	Meaning	Origin	Dimension
NUMBEX (20,20)	BLUE exogenous firepower unit identifiers	NODATFAC-Card 4-1,4...	20 x $NA
NUMBLO (20,20)	BLUE exogenous firepower nodes (location)	NODATFAC-Card 4-2,5...	$NA
BICEUN (20,20)	BLUE force parameter terms	NODATFAC-Card 4-3,6...	$NA
NUMREX (20,20)	RED exogenous firepower unit identifiers	NODATFAC-Card 5-1,4...	$NA
NUMRLO (20,20)	RED exogenous firepower nodes (location)	NODATFAC-Card 5-2,5...	$NA
RICEUN (20,20)	RED force parameter terms	NODATFAC-Card 5-3,6...	$NA
NBAUN (20)	Number of BLUE units in force parameter ratios	NODATFAC-7	$NA
NRAUN (20)	Number of RED units in force parameter ratios	NODATFAC-8	$NA
ITAN	Number of TIMEADV descriptors	--	---
ITNUN (40)	Number of units governed by TIMEADV instruction	TIMEADV-3	$TA
ITNDF (40)	Node for units to be located to receive instruction	TIMEADV-4	$TA
ITNDT (40)	Node to which units relocate	TIMEADV-5	$TA
TMADV (40)	Time of relocation	TIMEADV-7	$TA
ITUN (5,40)	Identifiers of units governed by TIMEADV	TIMEADV-8ff.	5 x $TA
ITRS (40)	Disengagement code	TIMEADV-6	$TA
IEXGN (2)	Number of units governed by EXOGUNIT card	EXOGUNIT-3	2
NEXGU (80,2)	Identifiers of units governed by EXOGUNIT card	EXOGUNIT-4ff.	$EX x 2 x 2
IOPX	Number of OPEXOGUN descriptors	--	---
IOPN (20)	Identifier of unit governed by OPEXOGUN card	OPEXOGUN-3	$OP
IOPJ (20)	Number of units taken under fire	OPEXOGUN-4	$OP
IOPUN (20,5)	Identifiers of units at which fire is directed	OPEXOGUN-5ff.	$OP x 5
IRRT (40)	Disengagement code for retracting units	RETREAT-7	$RT
IART (40)	Disengagement code for advancing unit	ADVANCE-7	$AD
NATRCD (20)	Force parameter ratio code	NODATFAC-3	$NA

Table C-7

COMMON BLOCK "F" ARRAYS

Parameter	Meaning	Origin	Dimension
IFRC	Number of FRCRATIO descriptors		
IFRCI (6)	OB = 1, LGST = 2, ICE = 3, DAY = 4	(FRCRATIO-3)	$FR
IFRCL (6)	Number of force ratio calculations to be made	(FRCRATIO-4)	$FR
FRMIN (6)	Minimum force level multiplier	(FRCRATIO-5)	$FR
FRMAX (6)	Maximum force level multiplier	(FRCRATIO-6)	$FR
FRDIF (6)	$(FRMAX_i - FRMIN_i)/(IFRCL_i - 1)$	Subroutine INPUT	$FR
IFRN (6)	Number of units to vary (neg. if RED)	(FRCRATIO-7)	$FR
IFRUN (30,6)	Identifiers of units governed by FRCRATIO card	FRCRATIO-8ff.	30 x $FR

Table C-8

COMMON BLOCK "FB" ARRAYS

Parameter	Meaning	Origin	Dimension
IFBM	Number of FEBA move cards ($\leq$ 3)	Subroutine INPUTB	1
NFBNOD (12,3)	Nodes for FEBA move calculations - in geographical order starting at RED end	FEBAMOVE-Card 1-7 to 19	-
NFBBN (3) NFBRN (3)	Number of BLUE/RED units that are to move w/FEBA ($\leq$ 20)	FEBAMOVE-Card-2 and 3	-
NFBBUN (20,3) NFBRUN (20,3)	Identifiers of units that move with the FEBA	FEBAMOVE-Card 2	-
NFBBNX (3) NFBRNX (3)	Number of BLUE/RED units supplying exogenous firepower to be directed at FEBAMOVE modes	FEBAMOVE-Card 1-4 and 5	-
NFBXUN (10,3) NFRXUN (10,3)	Identifiers of SPEC units which create exogenous fire-power directed at FEBAMOVE modes	FEBAMOVE-Card 3	-
FBBXF (10,3) FBRXF (10,3)	Scaling factor for exogenous firepower effectiveness of NFBXUN/NFRXUN units	FEBAMOVE-Card 3	-
NFBTAB (3)	Number of entries in FEBA movement rate table	FEBAMOVE-Card 1-6	-
TFBFR (12,3)	Casualty inflicting power ratio curve (X axis)	FEBAMOVE-Card 4	-
TFBMB (12,3)	End points for FEBA movement rate curve (Y axis)	FEBAMOVE-Card 4	-
IMPC	Number of parameter change cards	Subroutine INPUTB	1
NPMCD (100)	Code for variable to be changed (see page 3-23)	PARMCHNG-3	$PC

Table C-8 (Concluded)

Parameter	Meaning	Origin	Dimension
NPMUN (100)	Unit identifier for unit whose parameter is to be changed	PARMCHNG-4	$PC
TPMC (100)	Time at which parameter is to change	PARMCHNG-5	$PC
PMCQN (100)	Now or incremental parameter value	PARMCHNG-6	$PC

Table C-9

COMMON BLOCK "FBA" ARRAYS

Parameter	Meaning	Origin	Dimension
BMOV (3)	Furthest advance of BLUE toward RED	Subroutine ADCP6	3
BDAY (3)	Time of furthest advance BMOV	Subroutine ADCP6	3
RMOV (3)	Furthest advance of RED toward BLUE	Subroutine ADCP6	3
RDAY (3)	Time of furthest advance RMOV	Subroutine ADCP6	3
FPOS (3)	Instantaneous FEBA location	Subroutine ADCP6	3

Table C-10

COMMON BLOCK "G' ARRAYS

Parameter	Meaning	Origin	Dimension
IPA	Number of PROASIGN descriptors		
ICHA	Number of CHASE descriptors		
IRDV	Number of RENDEVOU descriptors		
ISTR	Number of STRTEGRT descriptors		
IPAJ (20)	Number of units in a PROASIGN set (neg. if RED)	PROASIGN-4	$PR
IPAU (7,20)	Identifiers of units in PROASIGN set	PROASIGN-5 through 11	7 x $PR
PAF (7,20,5)	Apportionment factor	PROASIGN-12 through 18	7 x $PR x 5
TPAF (20,4)	Time of reapportionment	PROASIGN-Card 2-1	$PR x 4
ICHAU (20)	Number of CHASE units in Set A (neg. if RED)	CHASE-3	$CH
ICHEU (20)	Number of CHASE units in Set B	CHASE-4	$CH

Table C-10 (Concluded)

Parameter	Meaning	Origin	Dimension
ICHIND (20)	CHASE criterion code ($\leq$ 5)	CHASE-6	$CH
ICHIN (20)	Number of CHASE units in Set C	CHASE-5	$CH
CHRAT (20)	Node for CHASE criterion	CHASE-7	$CH
ICHUN (20,20)	Identifiers of CHASE units in Set A	CHASE-8ff.-a	$CH x 20
ICHEUN (20,10)	Identifiers of CHASE units in Set B	CHASE-8ff.-b	$CH x 10
ICHIUN (20,10)	Identifiers of CHASE units in Set C	CHASE-8ff.-c	$CH x 10
IRDVU (20)	Number of RENDEVOU units in Set A	RENDEVOU-3	$RV
IRDVAU (20)	Number of RENDEVOU units in Set B	RENDEVOU-4	$RV
IRDVID (20)	RENDEVOU criterion code ($\leq$ 5)	RENDEVOU-6	$RV
IRDVIN (20)	Number of RENDEVOU units in Set C	RENDEVOU-5	$RV
RDVRAT (20)	Node for RENDEVOU criterion	RENDEVOU-7	$RV
IRDVUN (20,20)	Identifiers of RENDEVOU units in Set A	RENDEVOU-8ff.-a	$RV x 20
IRDVAN (20,10)	Identifiers of RENDEVOU units in Set B	RENDEVOU-8ff.-b	$RV x 10
IRVIUN (20,10)	Identifiers of RENDEVOU units in Set C	RENDEVOU-8ff.-c	$RV x 10
ISTRJ (40)	Number of units involved in withdrawal (neg. if RED)	STRTEGRT-3	$SR
ISTRFM (40)	Node from which units withdraw	STRTEGRT-4	$SR
ISTRAV (40)	Node to which units withdraw	STRTEGRT-5	$SR
STRAT (40)	Withdrawal trigger ratio	STRTEGRT-6	$SR
NSTRUN (20,40)	Units involved in withdrawal	STRTEGRT-7ff.	20 x $SR

Table C-11

COMMON BLOCK "H" ARRAYS

Parameter	Meaning	Origin	Dimension
IDRC	Number of DSTRIBUT descriptors		
IDRCL (12)	Number of force allocations to be made	DSTRIBUT-3	$DR
DRMIN (12)	Minimum fraction of forces to be allocated	DSTRIBUT-4	$DR
DRMAX (12)	Maximum fraction of forces to be allocated	DSTRIBUT-5	$DR
DRDIF (12)	$(DRMAX_i - DRMIN_i)/(IDRCL_i - 1)$	Subroutine INPUT	$DR
IDRUN (20,12)	Identifiers of units governed by DSTRIBUT card	DSTRIBUT-7ff.	20 x $DR
IDRN (12)	Number of units to vary (negative if RED)	DSTRIBUT-6	$DR

Table C-12

COMMON BLOCK "L" ARRAYS

Parameter	Meaning	Origin	Dimension
ILNODS	Number of BTLENODE - Nonhomogeneous card sets		
ILTN (10)	Nonhomogeneous battle node	BTLENODE-3	$BN
ILFCOD (10)	Surveillance capability code	BTLENODE-4	$BN
ILNBUN (10)	Number of BLUE units that can engage	BTLENODE-5	$BN
ILNRUN (10)	Number of RED units that can engage	BTLENODE-6	$BN
ILBUN (10,10)	BLUE unit identifiers in battle	BTLENODE-7ff.	10 x $BN
ILRUN (10,10)	RED unit identifiers in battle	BTLENODE-7ff.	10 x $BN
XLRB (10,10) XLBR (10,10)	RED on BLUE or BLUE on RED exogenous firepower effectiveness	Subroutine INPUT & ADCP	10 x $BN
ELBR (10,10,10)	Unit attrition coefficient BLUE on RED	BTLENODE-Card 2	10 x 10 x $BN
ELRB (10,10,10)	Unit attrition coefficient RED on BLUE	BTLENODE-Card 2	10 x 10 x $BN
SLBR (10,10,10) SLRB (10,10,10)	Surveillance effect of BLUE on RED or RED on BLUE	Subroutine INPUT & ADCP	10 x 10 x $BN
DTLW (10)	Weighted mean of delta time to reach defeat criterion	Subroutine INLVSV	$BN
ILXN	Number of exogenous firepower units	Subroutine INPUT	1
ILXUN (20)	Exogenous firepower unit identifier	BTLENODE-Card 3-2	$BN2
ILXS (20)	BTLENODE--Nonhomogenous card number for exogenous firepower	Subroutine INPUT	$BN2
ILXU (20)	Target unit identifier	BTLENODE-Card 3-3	$BN2
ILSN	Number of surveillance units	Subroutine INPUT	1
ILSUN (20)	Surveillance unit identifier	BTLENODE-Card 3-2	$BN3
ILSS (20)	BTLENODE--Nonhomogeneous card number for surveillance	Subroutine INPUT	$BN3
ILSU (20)	Surveillance target unit identifier	BTLENODE-Card 3-3	$BN3
ILSV (20)	Surveillance beneficiary	BTLENODE-Card 3-4	$BN3
FLSU (20)	Effect of friendly surveillance	BTLENODE-Card 3-5	$BN3

Table C-13

COMMON BLOCK "LG" ARRAYS

Parameter	Meaning	Origin	Dimension
INDC	Number of LGINTDIC card sets		
NPNDC (5)	Identifier of the pipeline unit	LGINTDIC-5	$LG
NPNDD (5)	Pipeline reserve unit identifier	LGINTDIC-6	$LG
LNPUN (5)	Location of the pipeline unit	LGINTDIC-7	$LG
NODPPI (5,30)	Firepower degradation code (ICE or NODEPROP card)	LGINTDIC-Card 4-2,6...14	$LG x $LG3
QNPN (5)	Throughput capacity/pipeline unit in weight/time	LGINTDIC-9	$LG
NUNDC (5)	Number of units to be resupplied	LGINTDIC-4	$LG
NDCN (5)	Number of nodes at which resupply takes place	LGINTDIC-3	$LG
TPPSUP (5)	Time delay for resupply	LGINTDIC-8	$LG
NUNDCU (5,30)	Resupplied unit identifiers	LGINTDIC-Card 4-1,5,9,13	$LG x $LG3
FNUNDC (5,30)	Resupply factors (supplies/time/component)	LGINTDIC-Card 4-3,7...15	$LG x $LG3
DNUNDC (5,30)	Supplies required before deployment	LGINTIDC-Card 4-4,8...	$LG x $LG3
NDCNOD (5,10)	Resupply nodes	LGINTDIC-Card 2	$LG x $LG2
TMNPN (5,3)	Regeneration times	LGINTDIC-Card 3-1,3,5	$LG x 3
FMNPN (5,3)	Regeneration rates (pipeline units/time)	LGINTDIC-Card 3-2,4,6	$LG x 3

Table C-14

COMMON BLOCK "LSOLA" ARRAYS

Parameter	Meaning	Origin	Dimension
ILSOL	Flag set to 1 if LINSOLU card read in data	INPUTB	1
NLINUN	Number of units to have order of battle determined	LINSOLU-Card 1-3	1
NTELN	Number of entries in table of trial force levels	LINSOLU-Card 1-4	1
NSUMLN	Position of SUMMARY card set that lists units whose surviving components are counted in the outcome	LINSOLU-Card 1-5	1
SOLIN	The required value of BLUE-RED survivors--i.e., the required scenario outcome	LINSOLU-Card 1-6	1
NUNOB (10)	Unit idents whose order of battle is to be determined	LINSOLU-Card 1-7ff.	10
OBTBLN (10)	Values in table of trial force levels	LINSOLU-Card 2	10

72

73

Table C-15

COMMON BLOCK "NLOG" ARRAYS

Parameter	Meaning	Origin	Dimension
NLOGCP (10)	User supplied program logic indicator array	Program control card-7 to 16	10

Table C-16

COMMON BLOCK "PATCH" ARRAYS

Parameter	Meaning	Origin	Dimension
NBG	Number of BLUE ground units	Subroutine INPUT2	1
NBA	Number of BLUE air units	Subroutine INPUT2	1
NBN	Number of BLUE naval units	Subroutine INPUT2	1
NBS	Number of BLUE special units	Subroutine INPUT2	1
NRG	Number of RED ground units	Subroutine INPUT2	1
NRA	Number of RED air units	Subroutine INPUT2	1
NRN	Number of RED naval units	Subroutine INPUT2	1
NRS	Number of RED special units	Subroutine INPUT2	1

Table C-17

COMMON BLOCK "OL11" ARRAYS

Parameter	Meaning	Origin	Dimension
NIND	Code indicating use of NODH geography	Program control card-6	1
NTOD	Flag to prevent reading NODH data more than once	Subroutine INILT	1
NA	Number of nodes read from NODH file	Subroutine RDNND	1

Table C-18

COMMON BLOCK "OVL22" ARRAYS

Parameter	Meaning	Origin	Dimension
MGA	Maximum number of column elements in subroutine OUTMTR	Subroutine SURFGN	1
MGB	Maximum number of row elements in subroutine OUTMTR	Subroutine SURFGN	1
M (18)	Loop counter array for FRCRATIO and DSTRIBUT cycles (1-6 for FRCRATIO, 7-12 for DSTRIBUT)	Subroutine SURFGN	18
MA	Column index for summary table in subroutine OUTMTR	Subroutine SURFGN	1
MB	Row index for summary table in subroutine OUTMTR	Subroutine SURFGN	1

Table C-19

COMMON BLOCK "OVL23" ARRAY

Parameter	Meaning	Origin	Dimension
NSTP	Flag set to 1 in ADCP when any STOPBTLE criteria met	Subroutine ADCP	1

Table C-20

COMMON BLOCK "OVL245" ARRAYS

Parameter	Meaning	Origin	Dimension
T	Simulation current time	Subroutine TSMO	1
DT	Incremental time to next event	Subroutine TSMO	1
NTC	Event type of event associated with DT	Subroutine TSMO	1

Table C-21

COMMON BLOCK "PLAN56" ARRAYS

Parameter	Meaning	Origin	Dimension
AMOBD (3)	FEBA instantaneous velocity	Subroutine ADCP6	3
TIMED (3)	Time associated with AMOBD	Subroutine ADCP6	3
DIST (3)	Distance of FEBA between two modes	Subroutine ADCP6	3
IJPM	Index to next PARMCHNG card to be processed	Subroutine ADCP5	1

74

Table C-22

COMMON BLOCK "PLN" ARRAYS

Parameter	Meaning	Origin	Dimension
LTNDT (40)	Flag used to indicate when change of objective has been made for a TIMEADVN unit.	Subroutine ADCP	$TA
LADJ (40)	Flag used to indicate whether an ADVANCE descriptor has been processed.	Subroutine ADCP2	$AD
LREJ (40)	Flag used to indicate whether a RETREAT descriptor has been processed.	Subroutine ADCP2	$RT
LOBJ (40)	Flag used to indicate whether an OBJCTADV descriptor has been processed.	Subroutine ADCP3	$OA
NFR (20)	Flag used to indicate whether a NODATFAC descriptor has been processed.	Subroutine ADCP3	$NA
FRATIM (20)	Time at which a force ratio value specified on a NODATFAC descriptor is first achieved.	Subroutine ADCP3	$NA
ICHASE (20)	Flag used to indicate whether a CHASE descriptor has been processed.	Subroutine ADCP2	$CH
IRENDE (20)	Flag used to indicate whether a RENDEVOU descriptor has been processed.	Subroutine ADCP2	$RV
KRSTR (20)	Status indicator for a RENDEVOU unit.	Subroutine ADCP2	$RV
KRSTC (20)	Status indicator for a CHASE unit.	Subroutine ADCP2	$CH
IDEB (120)	Packed indicators for BLUE/RED redeployment units.	Subroutine ADCP	$NU
IDER (120	(Set to REDEPLOY Card Number)	Subroutine ADCP	$NU
NBRS (120)	BLUE retreat status indicator.	Subroutine NODOBJ	$NU
NRRS (120)	RED retreat status indicator.	Subroutine NODOBJ	$NU
CON1 (4,80)	Scratch array for OB fractions for computations of exogenous firepower (see also CONINP in Common Block E).	Subroutine ADCP4	4 x $NP

Table C-23

COMMON BLOCK "R" ARRAYS

Parameter	Meaning	Origin	Dimension
NRDM	Number of RANDMSEQ descriptors	--	--
NRCD (20)	Code for parameter to be randomized	RANDMSEQ-3	$RS
AMENR (20)	Mean value	RANDMSEQ-4	$RS
STDR (20)	Standard deviation	RANDMSEQ-5	$RS
NRNU (20)	Number of units governed by the RANDMSEQ card (neg. if RED)	RANDMSEQ-6	$RS
NRUNT (20,30)	Identifiers of units involved	RANDMSEQ-7ff.	$RS
NRSTOP	Number of scenario iterations	RANDMSEQ-Card 2-2	1
IRGITG	Initial random number	RANDMSEQ-Card 2-3	1

Table C-24

COMMON BLOCK "RED" ARRAYS

Parameter	Meaning	Origin	Dimension
RED (120)	Order of battle (number of units)	UNITSPEC-5	$NU
RFP (120)	Exogenous firepower effectiveness (components attritted per time)	UNITSPEC-6	$NU
RTOS (120)	Defeat criterion (in O.B. units)	UNITSPEC-7	$NU
RNAF (120)	Index of combat effectiveness (ICE)	UNITSPEC-8	$NU
RTFD (120)	Time of first deployment	UNITSPEC-9	$NU
RMF (120)	Mobility factor	UNITSPEC-10	$NU
* RLOC (120)	Node location	Subroutine TSMO	$NU
* ROBJT (120)	Node objective (ultimate)	UNITSPEC-12	$NU
* ROBJ (120)	Node objective (immediate)	Subroutine TSMO	$NU
* RST (120)	Status	Subroutines TSMO, NODOBJ	$NU
RTA (120)	Time of last discontinuous event	Subroutine TSMO	$NU
RRNF (120)	Square root of normalized attrition factor	Subroutine MINV	$NU
RTLF (120)	Not used		$NU
* RPLF (120)	Location of applied exogenous support	UNITSPEC-13	$NU
* RDLC (120)	Initial node location	UNITSPEC-11	$NU

76

Table C-24 (Concluded)

Parameter	Meaning	Origin	Dimension
* RRST (120)	Retreat status	GUERILLA-4ff.	$NU
RILF (120)	Cumulative attrition inflicted by unit	Subroutine BATTLE	$NU
A4 TYPER (120)	Unit type (AIR, GRND, NAVL, SPEC)	UNITSPEC-3	$NU
RTH (120)	Survival ratio)(fraction 0.0 to 1.0)	UNITSPEC-7	$NU
RUE (120)	Unit effectiveness	Subroutine MINV	$NU
RFPF (120)	P Factor		$NU

*Denotes integer variable having "REAL" name.

A4 Denotes a 4 character per word holerith variable.

Table C-25

COMMON BLOCK "SUMRY" ARRAYS

Parameter	Meaning	Origin	Dimension
ISRY	Number of summary cards		1
-ISMTYP (40)	Summary type	SUMMARY-3	$SM
ISUMN (40)	Number of units included in summary	SUMMARY-4	$SM
ISUMUN (50,40)	Identifiers of units to be included in summary	SUMMARY-5ff.	50 x $SM
SMTL (10,40)	Title of summary (40 characters per title)	SUMMARY-Card 2	10 x $SM
SUMTA (28)	Summary matrix column headings	SUMTITLE-Card 1	28
SUMTD (4,11)	Summary matrix row headings	SUMTITLE-Card 2	4 x 11

Table C-26

COMMON BLOCK "SUNIT" ARRAYS

Parameter	Meaning	Origin	Dimension
ISMU	Number of SUMUNIT cards		1
-NSMUN (10)	Unit to which aggregated parameter is to be assigned	SUMUNIT-3	$SU
-NSMUC (10)	Type of aggregation code (1 to 8)	SUMUNIT-4	$SU
-NSMSUN (10)	Number of units to be aggregated (neg. if RED)	SUMUNIT-5	$SU
-NSMUNT (10,10)	Identifiers of all units to be aggregated	(SUMUNIT-6ff.)	$SU x NSMSUN

Table C-27

COMMON BLOCK "TEMP" ARRAYS

Parameter	Meaning	Origin	Dimension
IOPT (10)	Option flags for controlling selective processes	Program control card field 17 to 26	10
FARRAY (50)	Floating array for debugging purposes	--	50
IARRAY (50)	Fixed array for debugging purposes	--	50
TIT (20)	Title information from title card (80 characters)	Title card	20

78

Precedɪɴɢ Page Bʟank - ɴot Fɪʟmed

Appendix D

BALFRAM DESCRIPTOR OVERVIEW

11

Precedꞵing Page Blank - NOT FILMED

Appendix D

BALFRAM DESCRIPTOR OVERVIEW

1. INTRODUCTION

Input data for the BALFRAM program consist of descriptor cards and card sets. This appendix is included to provide the reader who is unfamiliar with BALFRAM a brief overview of the capabilities of each descriptor. For more detailed information readers are referred to the User Manual and the Seminar Guide.

For expository purposes, inputs to the BALFRAM program (descriptors) have been organized into five groups, as follows:

- Program control instructions
- Force definition inputs
- Battle logic inputs
- Movement logic inputs
- Sensitivity analysis instructions.

Each group of descriptors is discussed below.

2. PROGRAM CONTROL INSTRUCTIONS

These descriptors specify the number of nodes in the scenario, the type of output to be produced, and how that output is to be labeled.

2.1 Title Card

This card gives the scenario title.

2.2 Program Control Card

This card furnishes information to the program concerning: number of nodes used to generate geographic input, selection of attrition computation equations, maximum length of simulated battle, output of detailed battle history, elapsed time before battle history output is started, source of geographic input, and other options for processing.

2.3 LINSOLU Card

This is an optional program instruction that can be used to generate the order of battle or number of forces required to produce a desired battle outcome where the desired battle outcome is stipulated in terms of the difference between surviving BLUE and RED forces.

2.4 SUMMARY Card Set

This card set selects and labels information to be included in the battle summary output. Use of this card set is optional, but if no SUMMARY card set is input, there will be no summary printout.

2.5 SUMTITLE Card Set

This card set provides row and column headings when battle summary printout is in matrix form.

2.6 END Card

This card signals the end of the descriptor set.

3. FORCE DEFINITION INPUTS

These descriptors specify the sizes and capabilities of forces and include the option of changing force characteristics at user-specified times in the battle.

3.1 UNITSPEC Card

This card describes the size and capabilities of forces involved in the scenario.

3.2 GUERILLA Card

This card stipulates the conditions under which the various units may disengage when they are involved in a battle. If no specific disengagement rule is specified for a unit, it cannot disengage while involved in a battle.

3.3 PARMCHNG Card

This card (and its variants PVCHANGE, UVCHANGE, and UNCHANGE) changes the value of certain force characteristics at specified times during the scenario.

4. BATTLE LOGIC INPUTS

These descriptors specify where battles can take place and define additional force characteristics that are specific to conditions at those locations.

4.1 BTLENODE--Homogeneous Card Set

This card set describes the nodes at which homogeneous battles may occur. Homogeneous battles are interactions between sets of units of like type, e.g., BLUE ground forces versus RED ground forces. Additional information concerning attrition characteristics for BLUE and RED and the manner in which the attrition is to be calculated are also input with this card set.

4.2 BTLENODE--Nonhomogeneous Card Set

This card set describes nodes at which nonhomogeneous battles may occur. These are interactions between sets of units which are not necessarily of like type and whose mutual attrition can be approximated by the differential linear law.

4.3 STOPBTLE Card

This card causes battles to cease when a set of designated units fall below their respective defeat criteria, when any unit of a set reaches its objective, or when a time limit is exceeded.

4.4 NODATFAC Card Set

This card set inputs a tabular set of ICE adjustment factors that apply under stipulated conditions at a specified node. This card set would be used when the attrition factors for BLUE and RED input on the corresponding BTLENODE card do not pertain to all situations that may occur at that node.

4.5 NODEPROP Card

This card describes interaction between units of unlike type through exogenous firepower and specifies the effectiveness of exogenous firepower. It may also be used to substitute expected value computations for other programmed attrition computations.

4.6 EXOGUNIT Card

This card directs the exogenous firepower of a unit at enemy units which share the firing unit's nodal location.

4.7 OPEXOGUN Card

This card directs exogenous firepower of a specified unit at the location of a designated enemy unit.

4.8 PROASIGN Card Set

This card set assigns or redistributes available forces among specified units.

4.9 LGINTDIC Card Set

This card set describes the logistic pipelines of the scenario and incorporates the effects of reduction of pipeline capacity as a result of interdiction. Provision is made for pipeline regeneration rates, deployment and resupply rate factors, and locations of nodes which are to be supplied.

5. MOVEMENT LOGIC INPUTS

These descriptors control the planned and contingent movements of units in the scenario. Contingent movements are determined by deployments and events that occur during the battles.

5.1 ADVANCE Card

This card moves units from one node to another contingent upon the arrival of designated forces at a third node.

5.2 RETREAT Card

This card moves units from one node to another, contingent upon designated forces no longer being at a third node.

5.3 OBJCTADV Card

This card relocates specified units when enemy forces at the designated unit's present node are defeated or depart. However, if enemy units subsequently occupy the node from which the units have advanced, the friendly units return to this node.

5.4 STRTEGRT Card

This card withdraws units when the casualty-inflicting power ratio of the enemy to friendly forces is unfavorable. "Unfavorable" is also a quantitatively defined user input.

5.5 TIMEADVN Card

This card relocates units at stipulated times.

5.6 CHASE Card

This card causes one force to track another. The CHASE instruction can be implemented as a function of the location of friendly or enemy forces, the destruction of friendly or enemy forces, or the ratio of the casualty-inflicting power of friendly to enemy units at a designated node.

5.7 RENDEVOU Card

This card is similar to the CHASE card in that it causes a sequential link-up of forces. However, unlike the CHASE card, which is used to track or link-up forces of opposing sides, the RENDEVOU card links forces of the same side. The criteria for implementing the RENDEVOU instruction are identical to those for the CHASE card.

5.8 REDEPLOY Card Set

This card set redeploys designated units after a battle has been won. It also includes a provision for holding redeployed forces at a staging area for retrofit prior to redeployment.

5.9 FEBAMOVE Card Set

This card set resets unit mobility factors on the basis of the force ratio of the two opposing forces to reflect alternate movement rates at the Forward Edge of Battle Area (FEBA) during the course of a battle.

6. SENSITIVITY ANALYSIS INSTRUCTIONS

These descriptors control the changes in scenario parameters that are made between automatic, repetitive iterations in order to provide output for use in sensitivity and game theoretic analyses.

84

6.1 FRCRATIO Card

This card multiplies the order of battle, the index of combat effectiveness, or the exogenous firepower of units by specified factors.

6.2 DSTRIBUT Card

This card allocates progressively larger percentages of forces from one group of units to another group of units.

6.3 RANDMSEQ Card Set

This card set generates parameter sensitivity studies and allows certain parameters to be input as a normal distribution.

Appendix E

TACTICAL PROBLEMS IN MAN-COMPUTER INTERFACE

Preceding Page Blank - Not Filmed

Appendix E

TACTICAL PROBLEMS IN MAN-COMPUTER INTERACTIVE MODELS*

1. INTRODUCTION

Just a short time ago, a paper of this nature would not be considered suitable for an audience composed primarily of senior-level experts in military gaming. The subjects discussed here would probably be classified as simply computer programming problems that could be solved by programmers or technicians. But experience has taught us otherwise. Indeed, very often the failure to solve a tactical problem can prevent the realization of important innovations or breakthroughs of more theoretical types. As a simple example, linear programming applications became possible only because of the availability of sophisticated computer hardware and software.

During this conference, speakers have been mentioning the need for man-computer interactive gaming, which is also called "man-in-the-loop" gaming. This paper addresses some tactical problems that seem to be common to a number of man-computer interactive gaming models. These tactical problems involve some nonmilitary man-computer models, because, from a modeling point-of-view, they are very similar to military gaming models.

The term "interactive gaming" means computer-based gaming in which the users or operators may interject their decisions on-line through a remote terminal in a cooperative manner with the computer. It is not required that all the input data be submitted to the computer at the outset of the game because a simulation run can take only a single path from start to finish. It is not necessary, then, that contingency logic be supplied at every decision node. For this process, the amount of input, which otherwise would be required in conventional gaming, can be greatly reduced. If each game process is viewed as a path in a decision tree, the nodes that would require data input are shown in Figure E-1 as solid circles. In a non-interative process, all the nodes would require input data based on the chance that any path might be taken.

87

---

*This appendix documents a paper given at the Theater-Level Gaming and Analysis Workshop, Leesburg, Virginia on 29 September 1977 by Dr. Paul L. Tuan. The Workshop was sponsored by the Office of Naval Research.

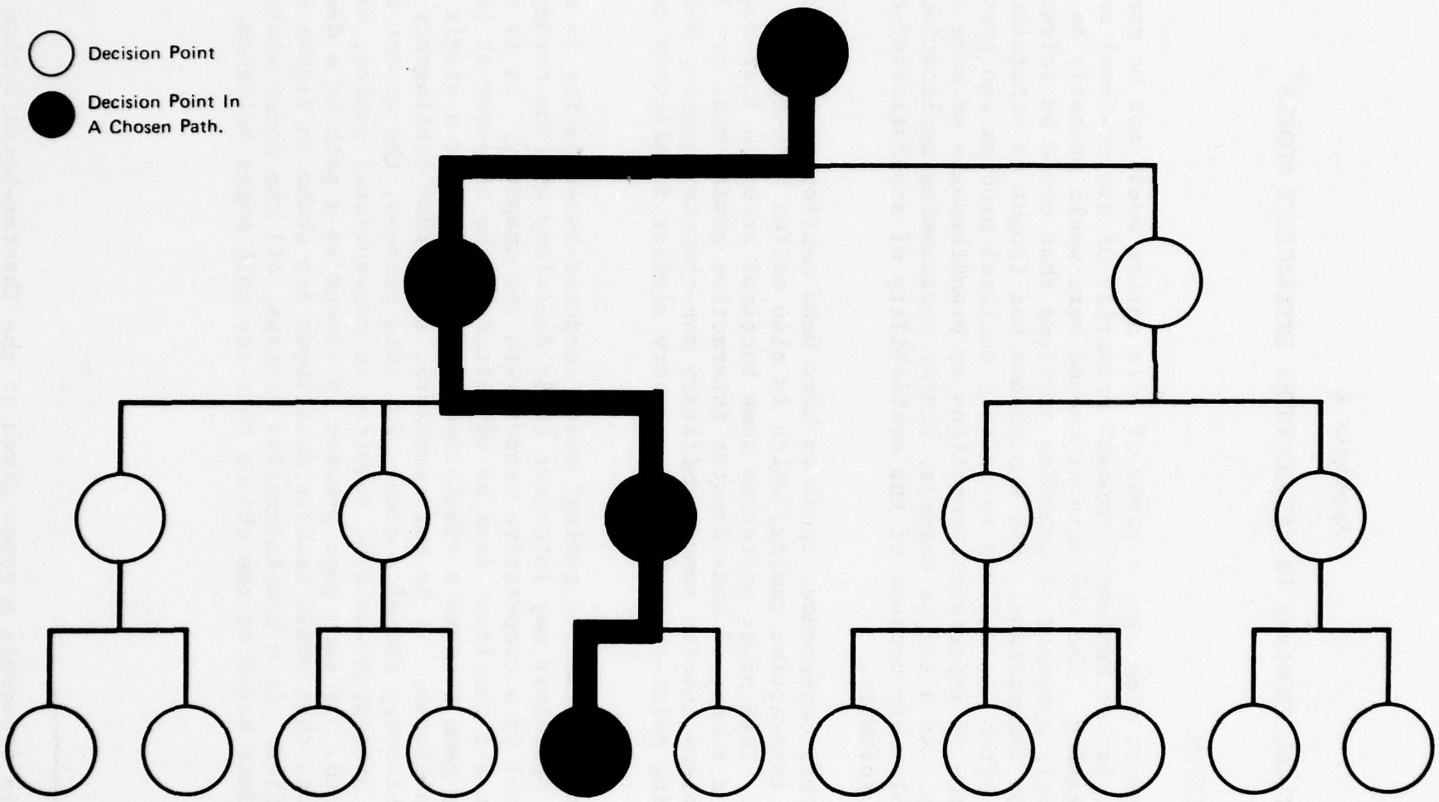


FIGURE E-1 A DECISION TREE SHOWING SINGLE PATH VERSUS MULTIPLE OUTCOMES

To use BALFRAM (Balanced Force Requirements Analysis Model) as an example, we envision a typical interactive process such as shown in Figure E-2. The points in the interactive simulation process of man-computer interface are highlighted. In general, at these points the man furnishes a list of data or simulation parameters used to control the next steps in the simulation.

The BALFRAM project is currently sponsored by the Fleet Analysis and Support Division (Code 230) of the Office of Naval Research, and we are now at the beginning of the second phase of interactive design. BALFRAM is currently used by CINCPAC and its subordinate commands as a theater-level gaming tool for analyzing JSOP or JSOP-related plans. Because of its flexibility in accepting user-designed contingency logic and different levels of data aggregation, BALFRAM has a built-in advantage for easy adaptation to man-computer interactive process.

In the course of designing and implementing an interactive type of system, often we encounter the following problems:

- Levels of human control
- Human judgment versus algorithmic judgment
- Decomposition schemes
- Computer information filtering and distribution
- Human factor problems.

These problems are not all inclusive, nor do they necessarily represent all of the most important problems in man-computer interactive design. Only a few significant problems are presented to show the type and scope of problems encountered.

2. LEVELS OF HUMAN CONTROL

Generally, there are two levels of human control that should be designed into an interactive process--the macrolevel and the microlevel. The macrolevel can also be called the "parametric level," and the micro-level can be called the "heuristics level." At the parametric level, the process remains essentially model-driven, but the operator may select and adjust certain parameter values or functions. For example, the operator may: change certain coefficient values in the objective function; select an alternative objective function; change on a probability function or its parameters; replace Lanchester equations with arbitrary functions. The influence of such control is global.

At the microlevel or heuristics level, the human operator may make specific decisions at certain decision points in a subjective manner. At this level, the operator may decide what percentage of the available air power should a specific unit commit to close air support (CAS) at a particular node or whether a red ground force unit should be considered viable for further combat. These decisions will be made by an on-line

AD-A051 832 SRI INTERNATIONAL MENLO PARK CA TRANSPORTATION AND I--ETC F/G 9/2
DEVELOPMENT OF AN INTERACTIVE BALFRAM COMPUTER PROGRAM: PHASE I--ETC(U)
NOV 77 B LAURENCE, P TUAN N00014-76-C-1111

UNCLASSIFIED SRI-TR-5822-1 NL

2 OF 2
AD A051 832

END
DATE FILMED
4 78
DDC

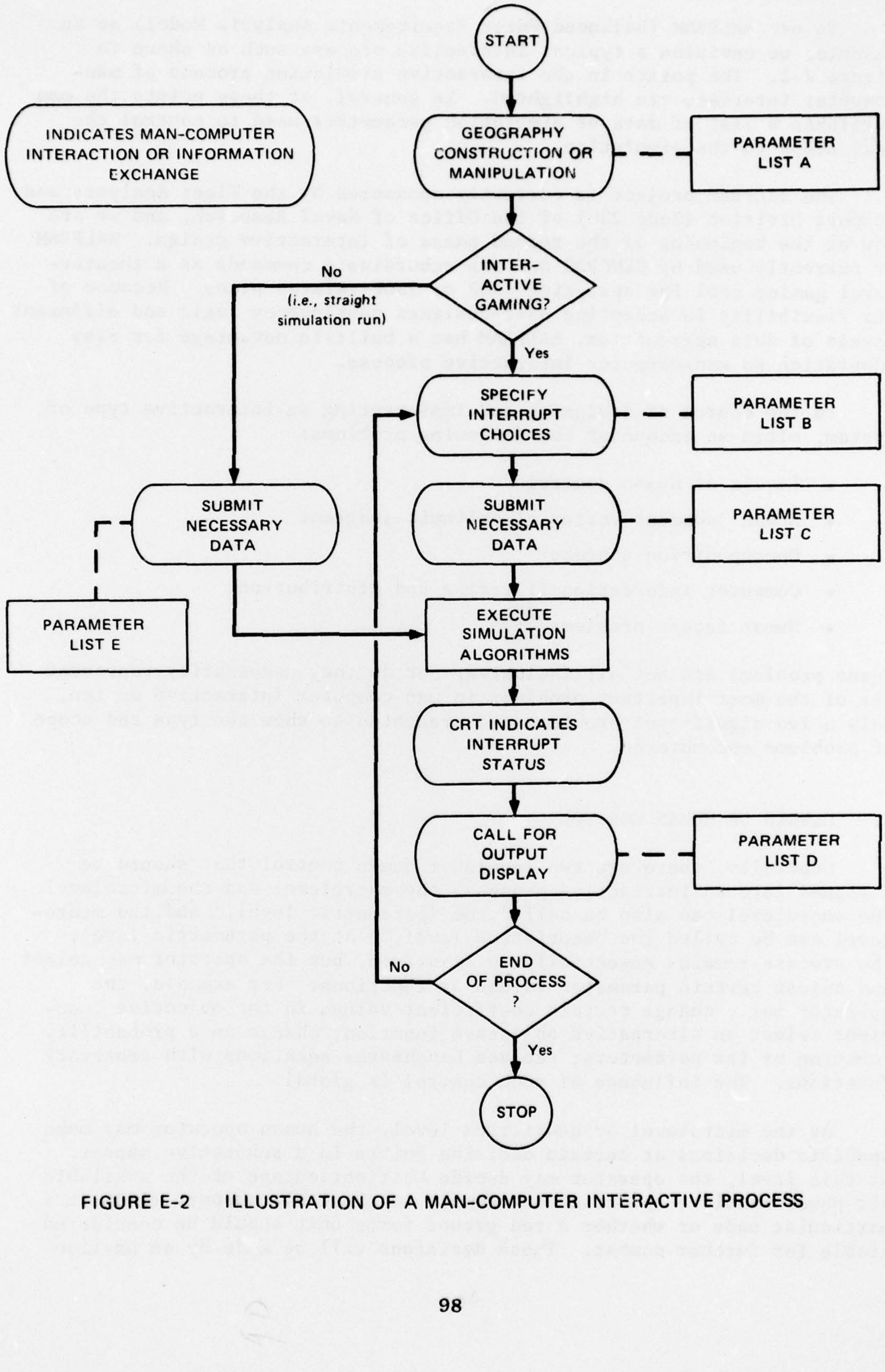


FIGURE E-2 ILLUSTRATION OF A MAN-COMPUTER INTERACTIVE PROCESS

evaluation of the battle conditions at the decision point rather than using predetermined decision criteria.

In designing and implementing this two-level approach, we have used two basic design methods. The first approach divides the macrolevels and microlevels into two independent stages as shown in Figure E-3. The first stage is designed for macrolevel control in which the operator may change control parameters and run the simulation on-line from the CRT iteratively. When no more improvements can be made at the macrolevel, then the operator will enter the next stage and make arbitrary and subjective decisions. The model, at this stage, would only serve as a referee and bookkeeper and would not attempt to conduct optimization procedures. Also, under this approach, once the process enters stage 2, it cannot return to stage 1 without forfeiting the changes made by the operator during stage 2.

The second approach is more flexible in mixing the macrolevels and microlevels. Figure E-4 depicts a system in which the normal control is automatic, but in which the operator can interrupt the decision loop at two points to make subjective decisions. This system is capable of returning to automatic model control after each interruption.

It is not clear that either of these approaches has a distinct advantage. The first approach is generally less costly to design because the two stages are segregated and no recycling is allowed. The second approach offers a greater flexibility for man-computer interaction.

3. *HUMAN JUDGMENT VERSUS ALGORITHMIC JUDGMENT*

There can be a problem when an operator must interface with a complex model that handles many parameters. Interface requirements must be specified and the operator decisions must be compatible with the algorithmic decisions (or vice versa). For example, if a model uses an interactive procedure for reaching an optimum, what would happen to the final solution when an intermediate value, which is a result of automatic computations, is replaced by a heuristically generated value? Would the computations become unstable? Part of the answer is in the proper design of filters and software controls so that the operator input cannot violate the designed algorithm objectives. Another part of the answer is in the method of decomposition, or partitioning of the model functions. If the interactive process is decomposed properly, the conflict between human judgment and algorithmic judgment can be avoided.

4. DECOMPOSITION SCHEMES

Unlike mathematical programming, the decomposition scheme in a man-computer interactive simulation process does not always lend itself to precisely defined terms. The most common means of partitioning an interactive process are by:

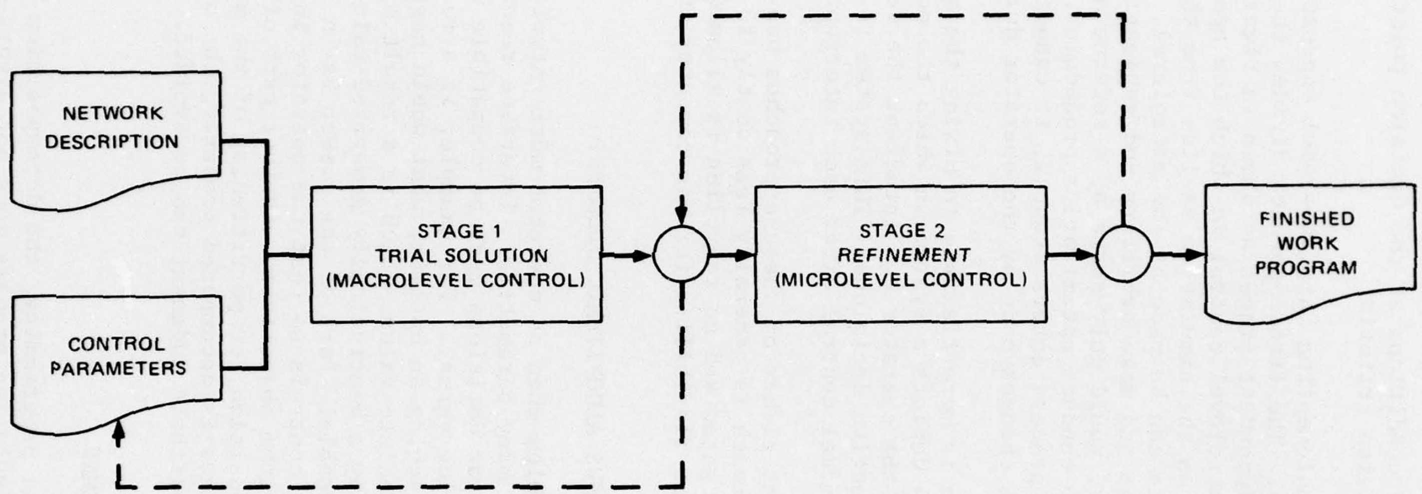


FIGURE E-3 INTERACTIVE PROCESS SEPARATING THE MACROLEVEL AND MICROLEVEL CONTROL

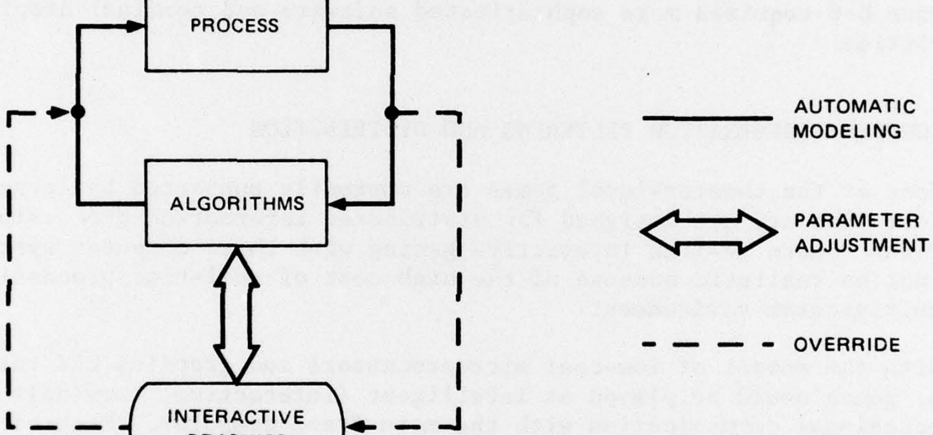


FIGURE E-4 INTERACTIVE PROCESS COMBINING THE MACROLEVEL AND MICROLEVEL CONTROLS

93

- Time division
- Geographical partition
- Hierarchical structure of decision-making
- Levels of aggregation.

Decomposition by geographical partition can be done easily as long as the resultant subgraphs are not interrelated in such a way that they have overlapping activities at the boundaries. Decomposition by levels-of-aggregation is a very worthwhile proposition because the model is more responsive to user requirements for data resolution and the user cost for data acquisition.

Our current effort on several man-computer projects is centered around a combination of time and decision hierarchy partitions. Figure E-5 shows a straightforward sequential simulation in which the gaming process within each stage can be executed without having to be concerned about other stages. Figure E-6 shows a more involved man-computer simulation in which the process can be recycled to previous stages. The method shown in Figure E-6 requires more sophisticated software and terminal display capabilities.

5. COMPUTER INFORMATION FILTERING AND DISTRIBUTION

Most of the theater-level games are currently supported by large computers that are not designed for distributed information processing. Performing remote on-line interactive gaming with these computer systems would not be realistic because of the high cost of real-time processing in a multiprogram environment.

With the advent of low-cost microprocessors and graphics CRT terminals, games could be played at intelligent (interactive) terminals with only occasional communication with the main-frame computer. Much of the map composition, input/output data editing and formatting, instructions to operators, and even some simple gaming logic can be done at the local terminal (see Figure E-7). The communication between the local terminal and the main-frame computer can be on a batch basis, which would significantly reduce processing cost. The determination of what and how much information is to be transmitted back and forth is a design task that should be studied in depth, then carefully specified.

6. HUMAN FACTOR PROBLEMS

Often the tasks of CRT formatting such as color selection (if color CRT is to be used), keyboard arrangement, and graphics generation, are left to the programmers as the last stage of the design phase. If an interactive tool is to be introduced to decision-makers the console environment must be designed to meet the needs of noncomputer personnel and not solely for professional programmers or operators. The problems of console management, reaction time, performance feedbacks, stress factors, human cognitive process, and CRT screen formatting, and the like, should have high priority design and implementation status.

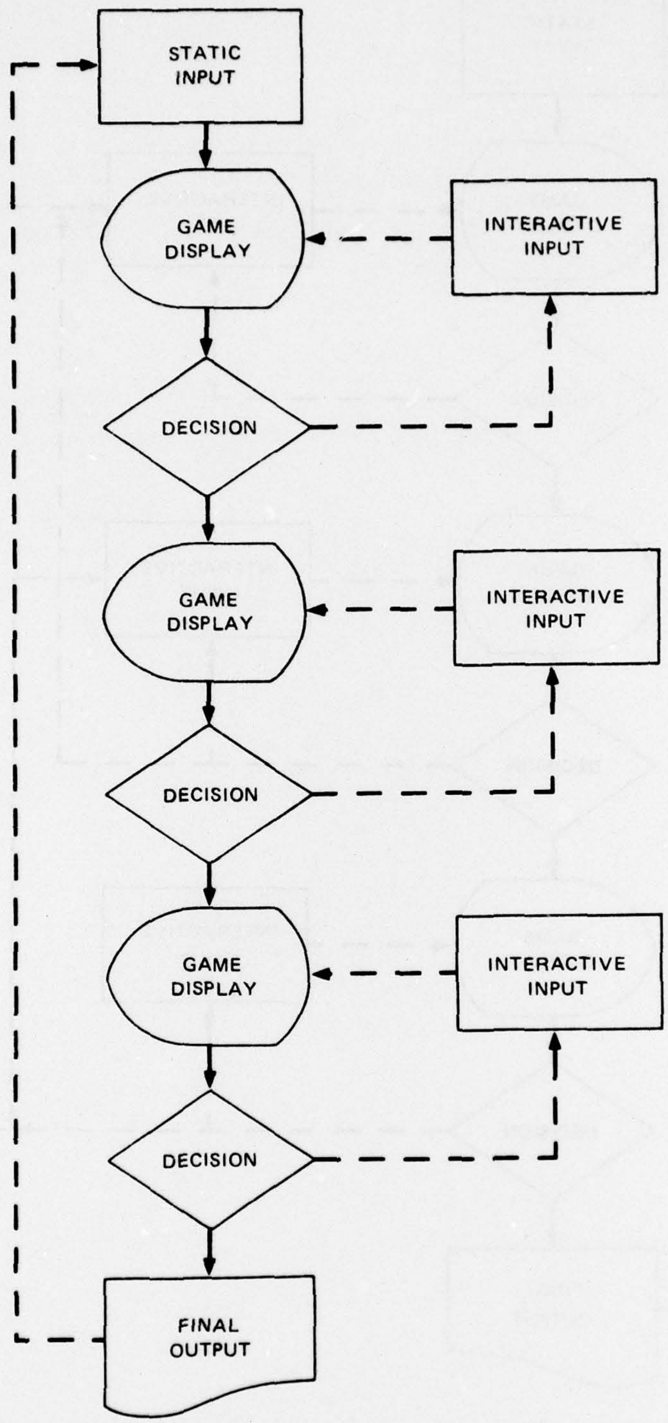


FIGURE E-5 SEQUENTIAL SIMULATION

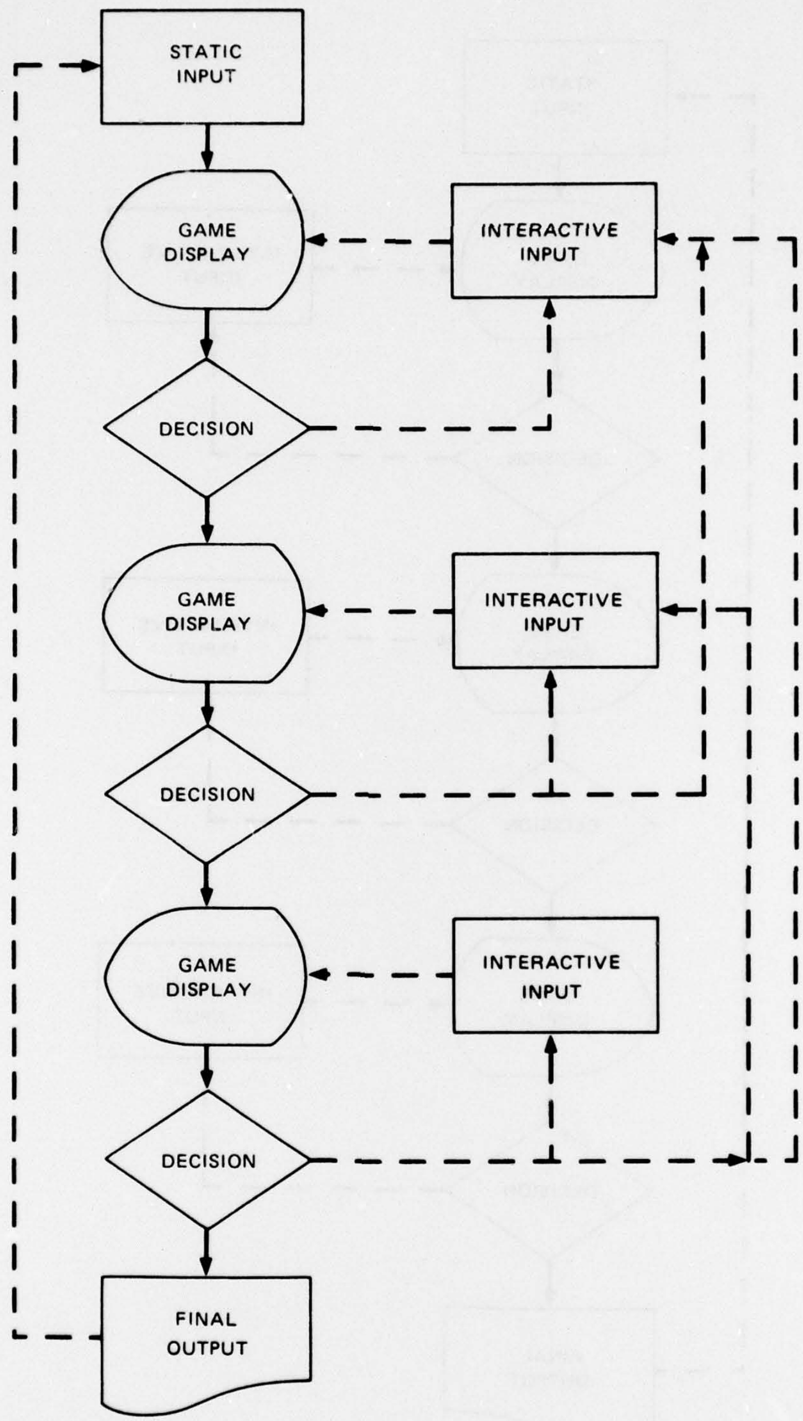


FIGURE E-6 RECYCLING SIMULATION

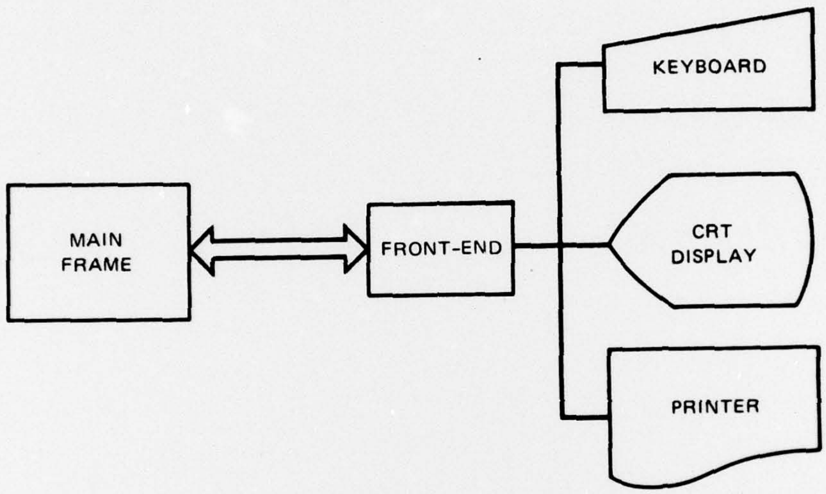


FIGURE E-7 A DISTRIBUTED SIMULATION PROCESS

97

Preceding Page Blank - NOT FILMED

DISTRIBUTION LIST

ORGANIZATION	NO. OF COPIES
Chief of Naval Operations Department of the Navy Washington, D.C. 20350	
Op-098T	1
Op-95	1
Op-97	1
Office of Naval Research Department of the Navy Arlington, Virginia 22217	
Code 431	2
Code 230	1
Commander in Chief, Pacific FPO, San Francisco, California 96610	
J77	5
J63	1
Commander in Chief United Nations Command, Korea APO, San Francisco 96301 (Attn: Mr. C. H. Phillips Science Advisor)	1
Commander in Chief Pacific Air Forces, (Honolulu-Oahu) APO, San Francisco 96553 (Attn: Dr. R. Hiller)	1
Commander, U.S. Taiwan Defense Command (J-3) (Taipei, Republic of China) APO, San Francisco 96263	1
Japan Defense Agency Operations Research Section	1